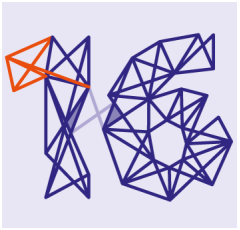
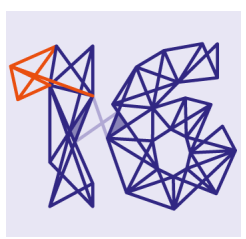


Clustering in networks

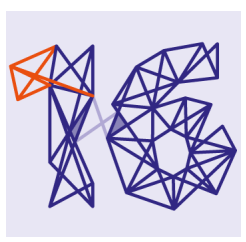
Vladimir Batagelj
IMFM Ljubljana, IAM UP Koper and NRU HSE Moscow

16th IFCS conference
Thessaloniki, August, 26–29, 2019



Outline

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
Current version of slides (August 28, 2019 at 23:55): slides PDF
https://github.com/bavla/biblio/blob/master/doc/SS/clustnet.pdf



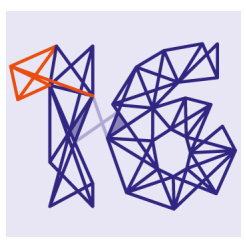
Networks

A *network* is based on two sets – a set of *nodes* (vertices), that represent the selected *units*, and a set of *links* (lines), that represent *ties* between units. They determine a *graph*. A link can be *directed* – an *arc*, or *undirected* – an *edge*.
Additional data about nodes or links may be known – their *properties* (attributes). For example: name/label, type, age, value, ...

Network = Graph + Data

A *network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ consists of:

- a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{A}$ is the set of arcs, $\mathcal{E}$ is the set of edges, and $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ is the set of links.
  $n = |\mathcal{V}|$, $m = |\mathcal{L}|$
- $\mathcal{P}$ *node value functions* / properties: $p : \mathcal{V} \to A$
- $\mathcal{W}$ *link value functions* / weights: $w : \mathcal{L} \to B$

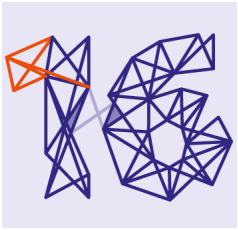


Large networks

The *size* of a network is usually measured by the number of nodes n and the number of links m. In a simple network (no parallel links) it holds $m \leq n^2$.

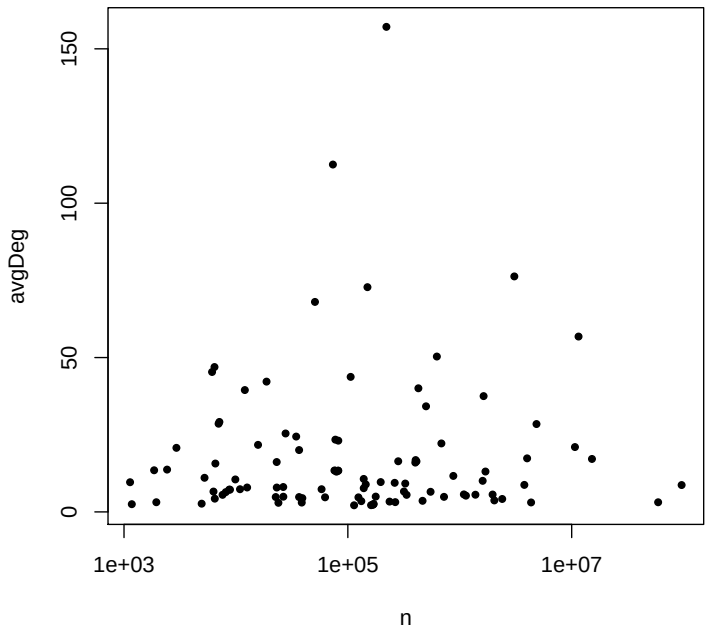
Large networks are networks with at least some thousands of nodes that can be stored entirely in the computer's memory. Huge networks.

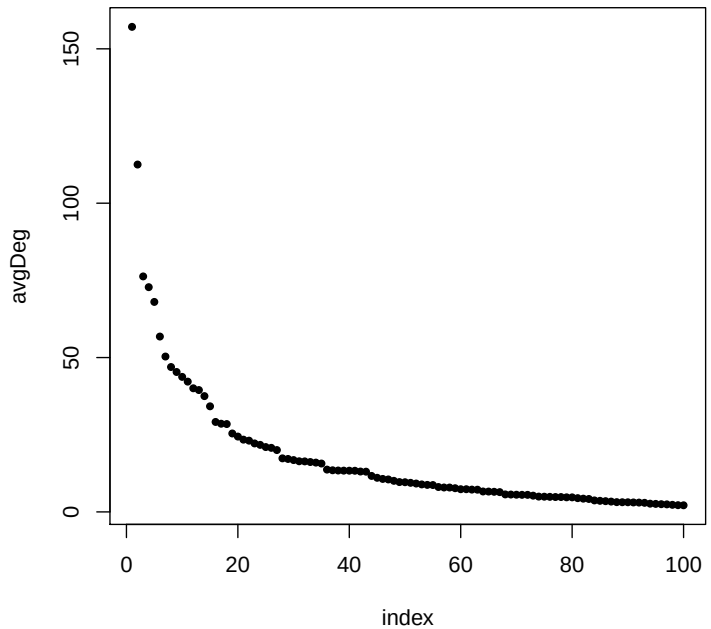
Large networks are usually *sparse*, $m \leq k \cdot n$, where $k << n$ (see Dunbar's number). This is a crucial property that often allows us to develop efficient (subquadratic) algorithms for analysis of large networks.



Dunbar's number

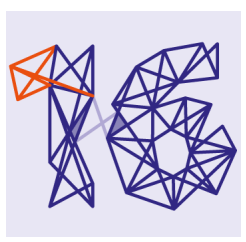
Average degrees of the SNAP and Konect networks





Average degree $\bar{d} = \frac{1}{n}\sum_{v \in V} \deg(v) = \frac{2m}{n}$. Most real-life large networks are *sparse* – the number of nodes and links are of the same order. This property is also known as a Dunbar's number.

The basic idea is that if each vertex has to spend for each link certain amount of "energy" to maintain the links to selected other vertices then, since it has a limited "energy" at its disposal, the number of links should be limited. In human networks the Dunbar's number is between 100 and 150.

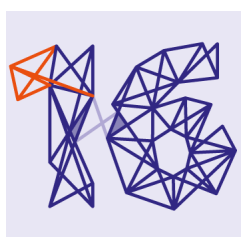


Complexity of algorithms

Let us look to time complexities of some typical algorithms:

	T(n)	1.000	10.000	100.000	1.000.000	10.000.000
LinAlg	O(n)	0.00 s	0.015 s	0.17 s	2.22 s	22.2 s
LogAlg	O($n \log n$)	0.00 s	0.06 s	0.98 s	14.4 s	2.8 m
SqrtAlg	O($n\sqrt{n}$)	0.01 s	0.32 s	10.0 s	5.27 m	2.78 h
SqrAlg	O(n^2)	0.07 s	7.50 s	12.5 m	20.8 h	86.8 d
CubAlg	O(n^3)	0.10 s	1.67 m	1.16 d	3.17 y	3.17 ky

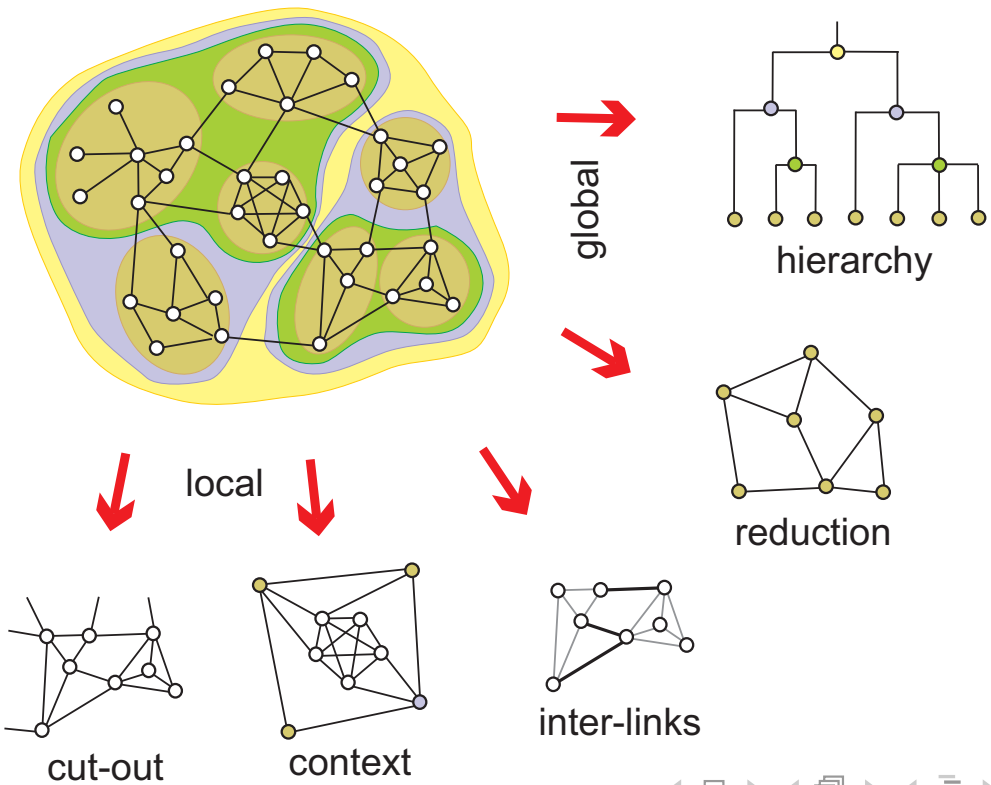
For the interactive use on large graphs already quadratic algorithms, O(n^2), are too slow.

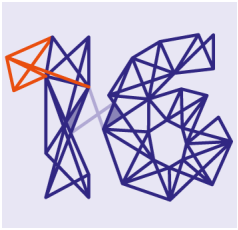


Approaches to large networks

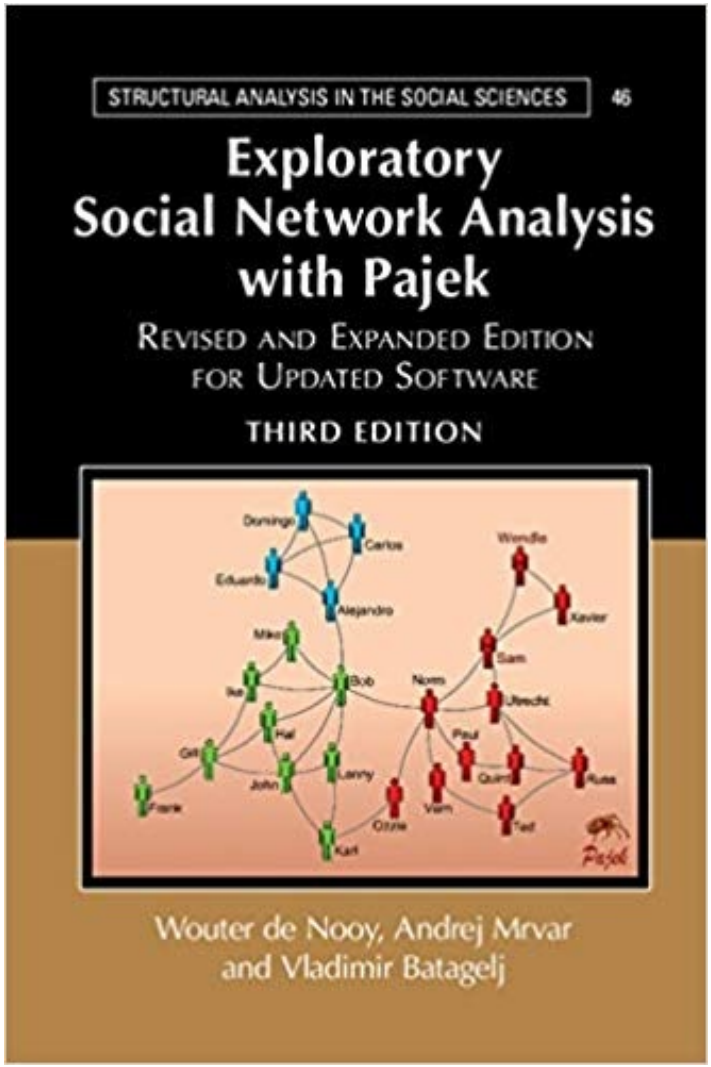
Most of *large* networks can not be displayed readable in their totality; also there are only few algorithms available for their analysis.

To analyze a large network we can use statistical approach or we can identify smaller (sub) networks that can be analyzed further using more sophisticated methods.





ESNA Pajek

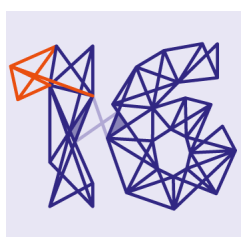


An introduction to social network analysis with Pajek is available in the book ESNA 3 (de Nooy, Mrvar, Batagelj, CUP 2005, 2011, 2018).

ESNA in Japanese was published by Tokyo Denki University Press in 2010; and in Chinese by Beijing World Publishing in November 2012.

Pajek – program for analysis and visualization of large networks is freely available, for noncommercial use, at its web site.

http://mrvar.fdv.uni-lj.si/pajek/

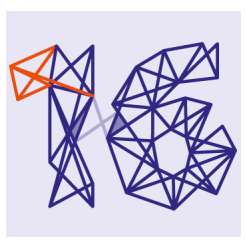


Networks and clustering

- clusters of elements (nodes or links) of a given network
  - selected (important) clusters
  - a complete clustering of all elements
- clustering of set of networks

Usually we are searching for disjoint clusters. Although in some applications the overlapping clusters are a natural solution.

In this talk we will deal only with the clusters of elements of a given network.



Selected important clusters – Cuts

The standard approach to find interesting groups inside a network was based on properties/weights – they can be *measured* or *computed* from network structure.

The *node-cut* of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \rightarrow \mathbb{R}$, at selected level t is a subnetwork $\mathcal{N}(t) = (\mathcal{V}', \mathcal{L}(\mathcal{V}'), p)$, determined by the set

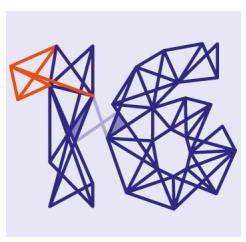
$$\mathcal{V}' = \{v \in \mathcal{V} : p(v) \geq t\}$$

and $\mathcal{L}(\mathcal{V}')$ is the set of links from $\mathcal{L}$ that have both endnodes in $\mathcal{V}'$.

The *link-cut* of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{L} \rightarrow \mathbb{R}$, at selected level t is a subnetwork $\mathcal{N}(t) = (\mathcal{V}(\mathcal{L}'), \mathcal{L}', w)$, determined by the set

$$\mathcal{L}' = \{e \in \mathcal{L} : w(e) \geq t\}$$

and $\mathcal{V}(\mathcal{L}')$ is the set of all endnodes of the links from $\mathcal{L}'$.

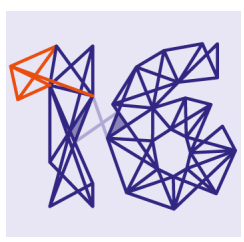


Selected important clusters – Cuts

From a cut we can get a partial clustering $\mathbf{C}(t)$ with connected components as clusters. For different thresholds, these clusterings are nested – they form a hierarchy.

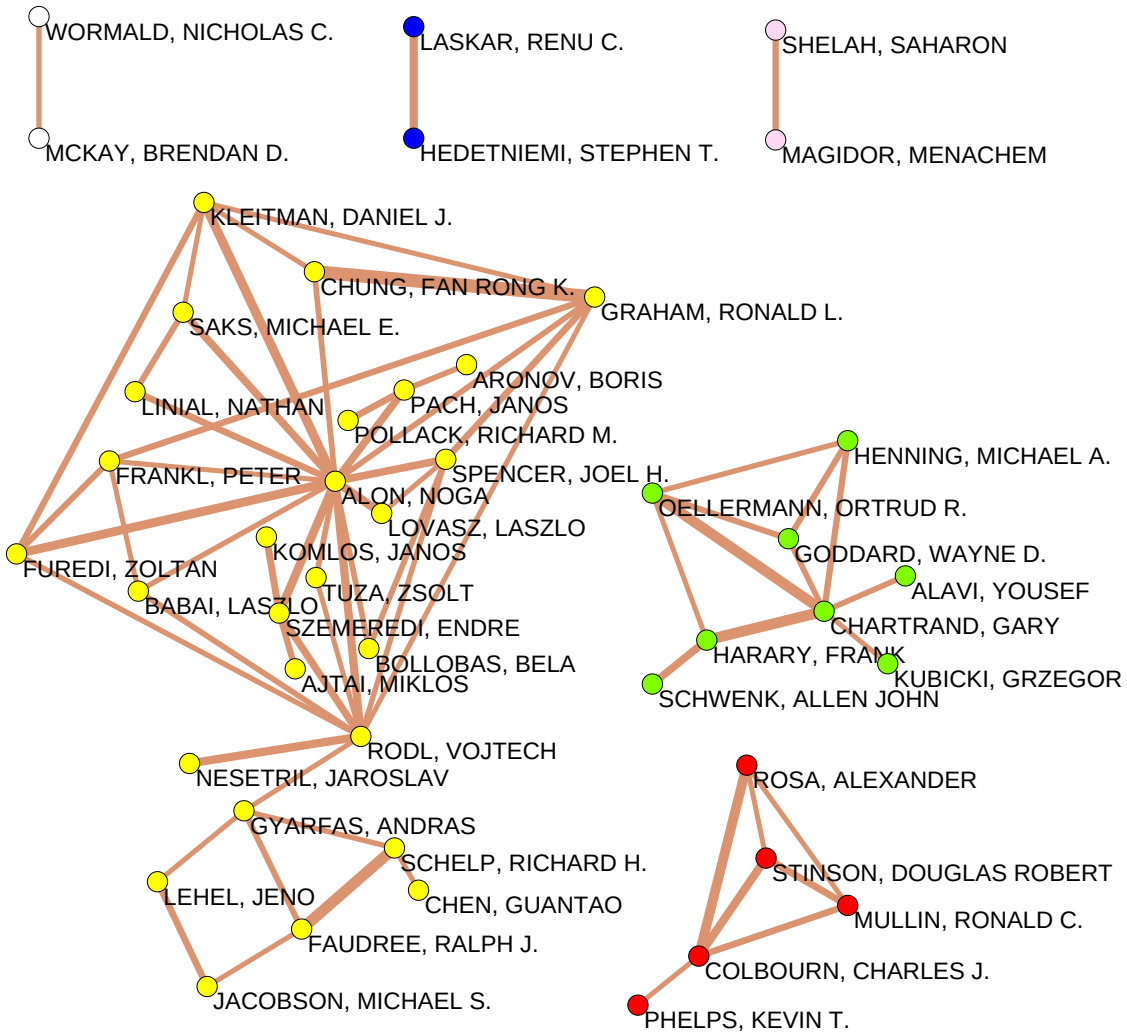
An elaborated version of cuts approach is provided with the *islands* approach [Batagelj et al.(2014), Subsection 2.9.1]. Islands also form a hierarchy for a selected node property of a given network.

Cuts and islands algorithms are subquadratic on sparse networks and can efficiently deal with large networks.

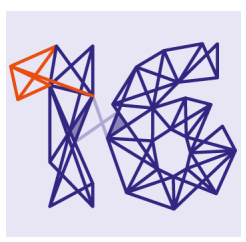


Edge-cut at level 16 of a triangular network

of Erdős collaboration graph

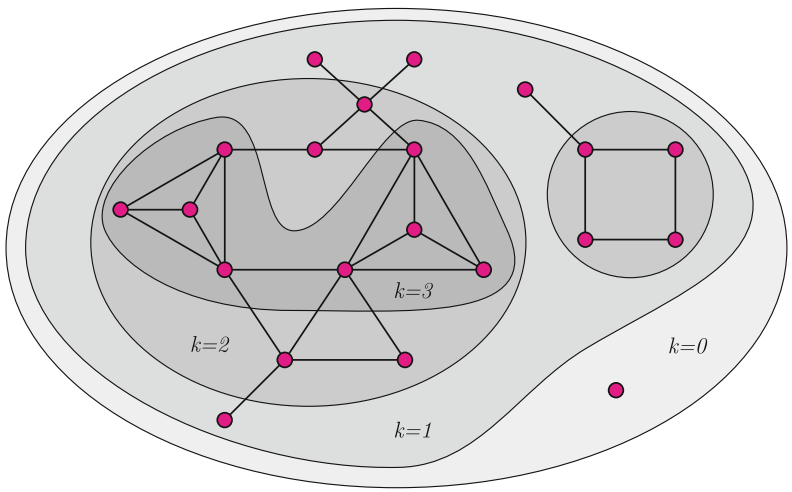


without Erdős,
$n = 6926$,
$m = 11343$



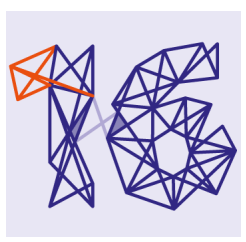
Cores and generalized cores

Cores are used to identify dense parts of a network.



The notion of core was introduced by Seidman in 1983.
Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph. A subgraph $\mathcal{H} = (C, \mathcal{E}|C)$ induced by the set C is a *k-core* or a *core of order* k iff $\forall v \in C : \deg_{\mathcal{H}}(v) \geq k$, and $\mathcal{H}$ is a maximal subgraph with this property. The core of maximum order is also called the *main* core.

The *core number* of a node v is the highest order of a core that contains this node. The degree $\deg(v)$ can be: in-degree, out-degree, in-degree + out-degree, etc., determining different types of cores.



Properties of cores

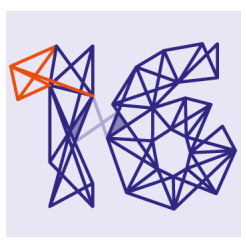
From the figure, representing 0, 1, 2 and 3 core, we can see the following properties of cores:

- The cores are nested: $i < j \implies \mathcal{H}_j \subseteq \mathcal{H}_i$
- Cores are not necessarily connected subgraphs.

An efficient algorithm for determining the cores hierarchy is based on the following property:

> *If from a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we recursively delete all nodes, and edges incident with them, of degree less than k, the remaining graph is the k-core.*

Cores can also be used to localize the search for interesting subnetworks in large networks since: if it exists, a k-component is contained in a k-core; and a k-clique is contained in a k-core.

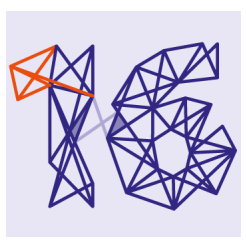


Generalized cores

The notion of core can be generalized to networks. Let $\mathcal{N} = (\mathcal{V}, \mathcal{E}, w)$ be a network, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph and $w : \mathcal{E} \rightarrow \mathbb{R}$ is a function assigning values to edges. A *node property function* on **N**, or a *p-function* for short, is a function $p(v, U)$, $v \in \mathcal{V}$, $U \subseteq \mathcal{V}$ with real values. Let $N_U(v) = N(v) \cap U$. Besides degrees and (corrected) clustering coefficient, here are some examples of p-functions:

$$
\begin{aligned}
p_S(v, U) &= \sum_{u \in N_U(v)} w(v, u), \text{ where } w : \mathcal{E} \rightarrow \mathbb{R}_0^+ \\
p_M(v, U) &= \max_{u \in N_U(v)} w(v, u), \text{ where } w : \mathcal{E} \rightarrow \mathbb{R} \\
p_k(v, U) &= \text{number of cycles of length } k \text{ through the node } v \text{ in } (U, \mathcal{E}|U)
\end{aligned}
$$

The subgraph $\mathcal{H} = (C, \mathcal{E}|C)$ induced by the set $C \subseteq \mathcal{V}$ is a *p-core at level* $t \in \mathbb{R}$ iff $\forall v \in C : t \leq p(v, C)$ and C is a maximal such set.



Generalized cores algorithm

The function p is *monotone* iff it has the property

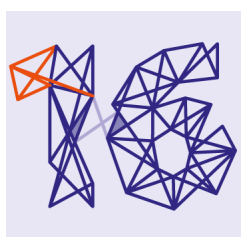
$$C_1 \subset C_2 \Rightarrow \forall v \in \mathcal{V} : (p(v, C_1) \leq p(v, C_2))$$

The degrees and the functions p_S, p_M and p_k are monotone. For a monotone function the p-core at level t can be determined, as in the ordinary case, by successively deleting nodes with value of p lower than t; and the cores on different levels are nested

$$t_1 < t_2 \Rightarrow \mathcal{H}_{t_2} \subseteq \mathcal{H}_{t_1}$$

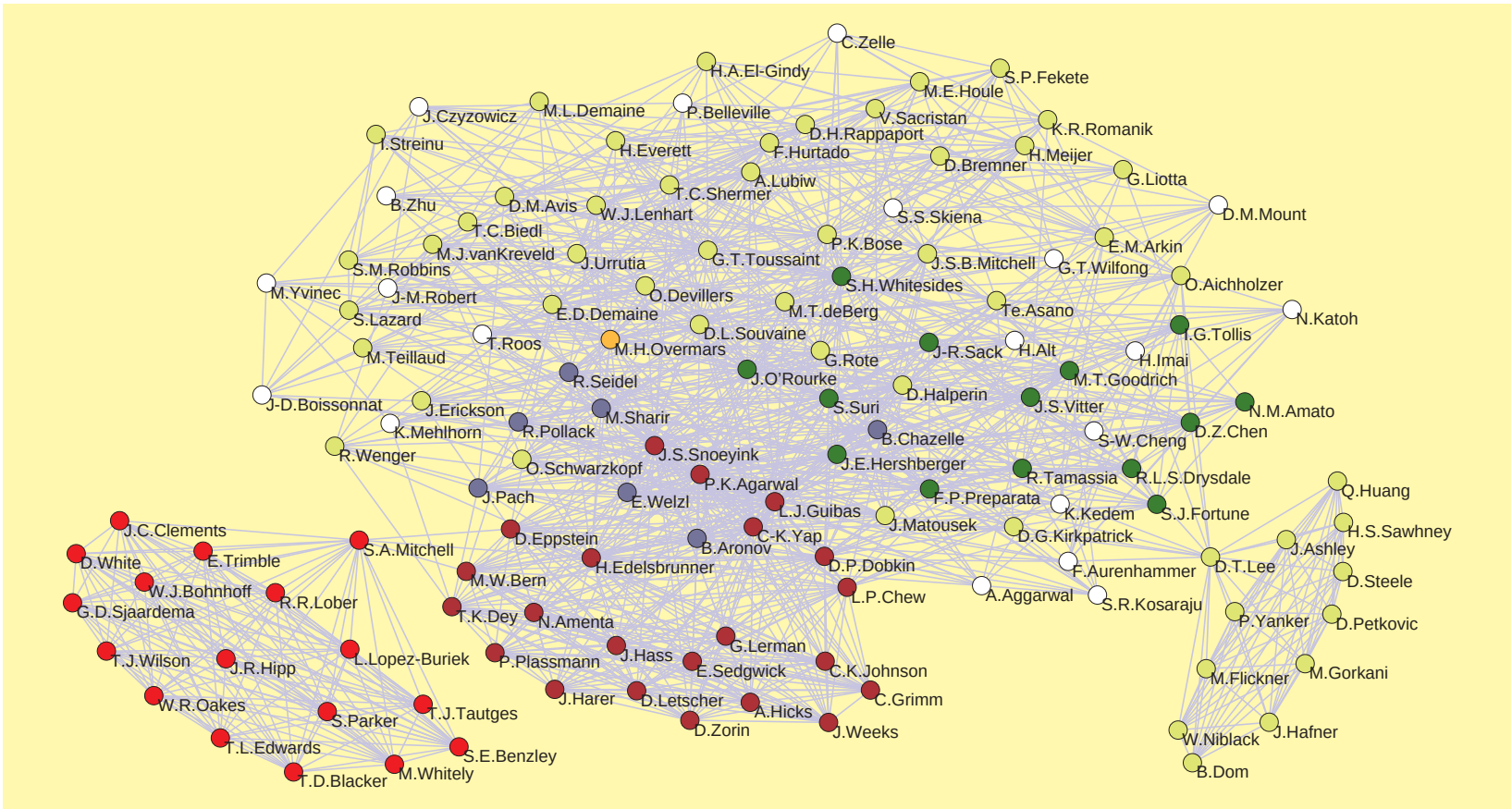
The p-function is *local* iff $p(v, U) = p(v, N_U(v))$.
The degrees, p_S and p_M are local; but p_k is **not** local for $k \geq 4$. For a local p-function an $O(m \max(\Delta, \log n))$ algorithm for determining the p-core levels exists, assuming that $p(v, N_C(v))$ can be computed in $O(\deg_C(v))$. paper

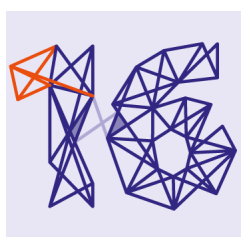
Extensions: two-mode networks, temporal networks. paper



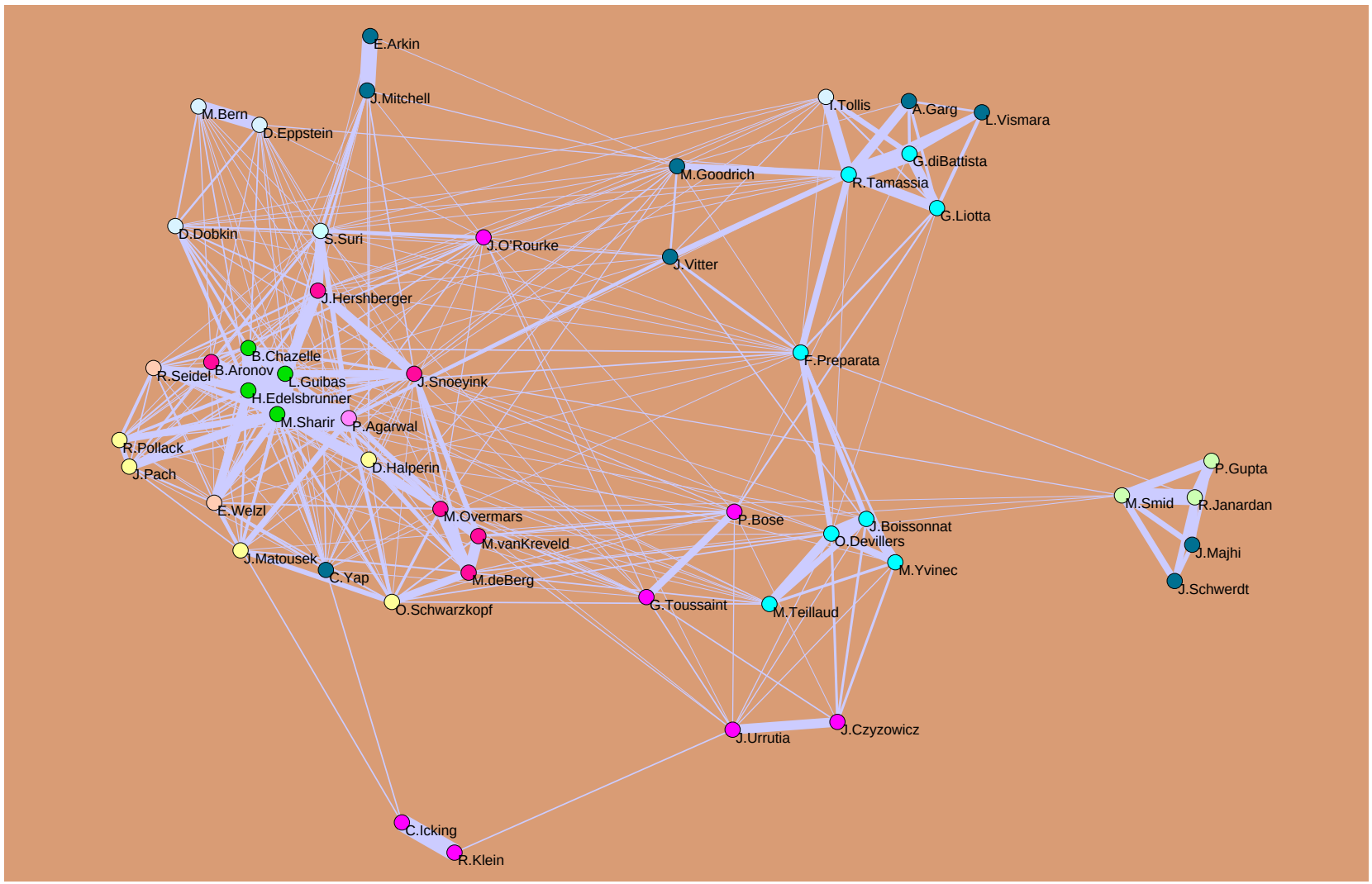
Cores of orders 10–21 in Computational Geometry

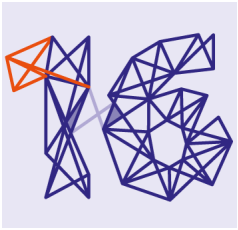
n=7343, M =11898





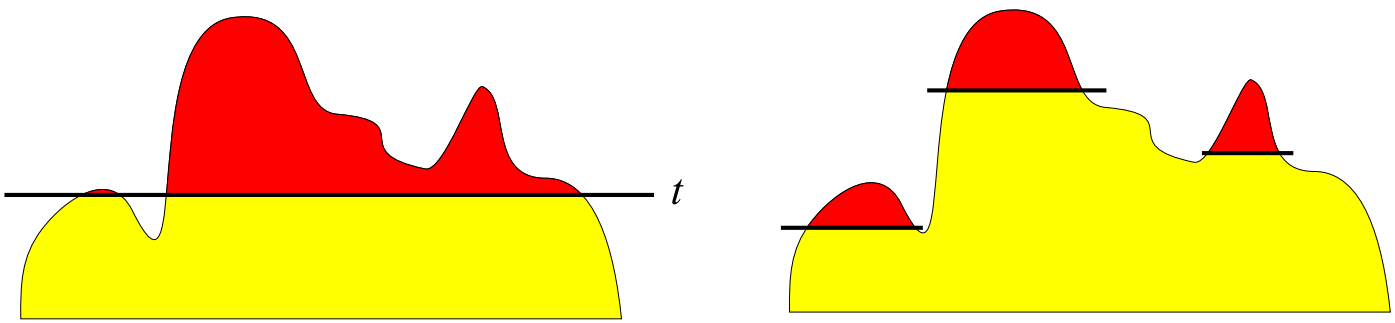
p_S-core at level 46 in Computational Geometry network



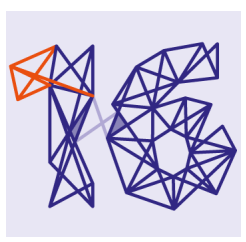


Islands

If we represent a selected property of nodes / links as a height of nodes / links and we immerse the network into a water up to selected level we get *islands*. Varying the level we get different islands.



We developed very efficient algorithms to determine the islands hierarchy and to list all the islands of selected sizes.
See details.



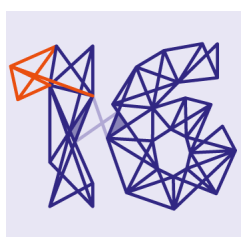
. . . Islands

Islands are very general and efficient approach to determine the 'important' subnetworks in a given network.

We have to express the goals of our analysis with a related property of the nodes or weight of the links. Using this property we determine the islands of an appropriate size (in the interval from k to K, $[k, K]$).

In large networks we can get many islands which we have to inspect individually and interpret their content.

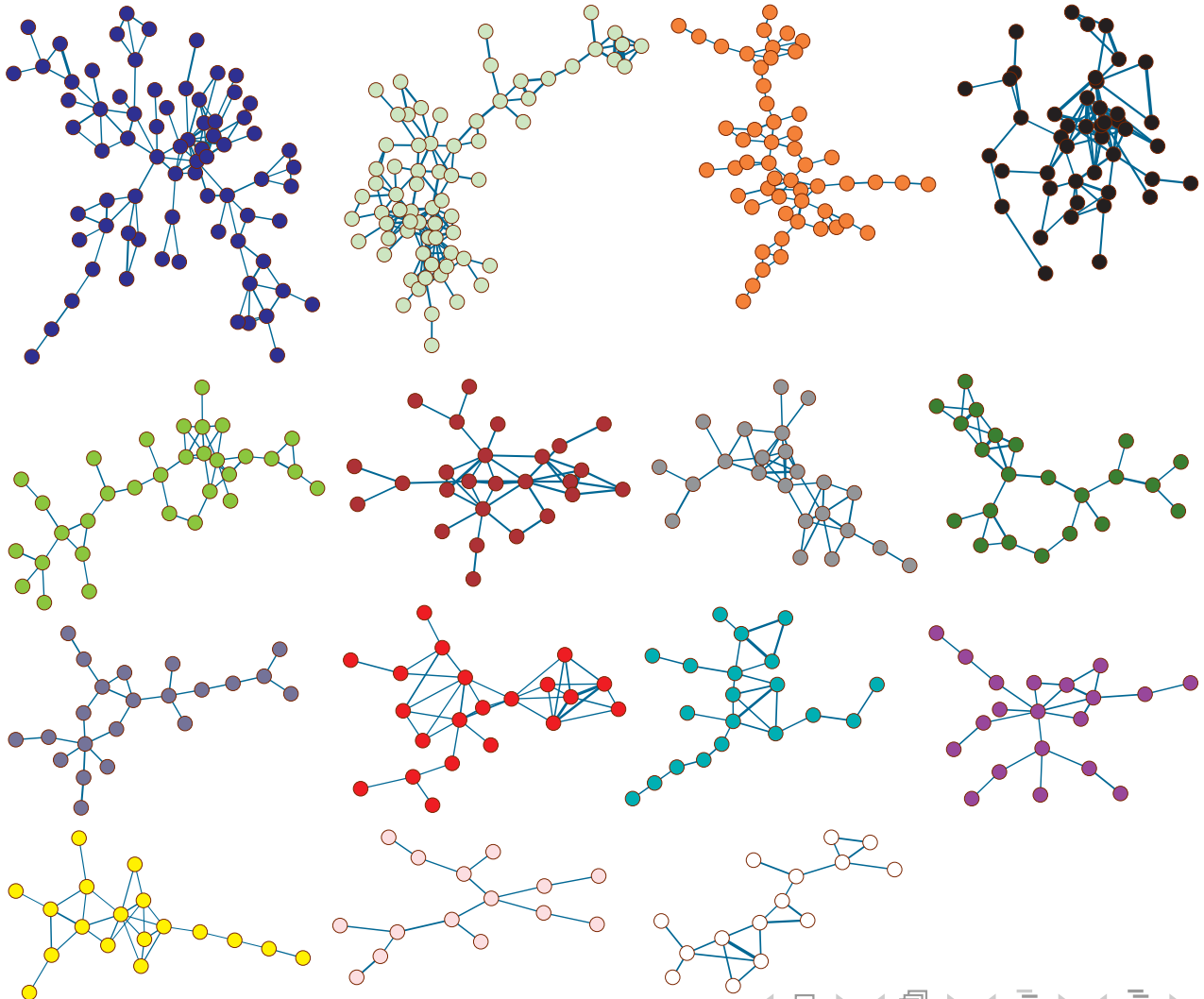
An important property of the islands is that they identify locally important subnetworks on different levels. Therefore they detect also emerging groups.

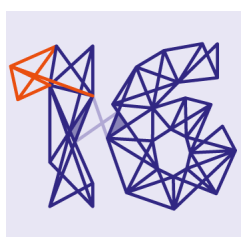


Bibliographic Coupling

Jaccard islands [15, 75]

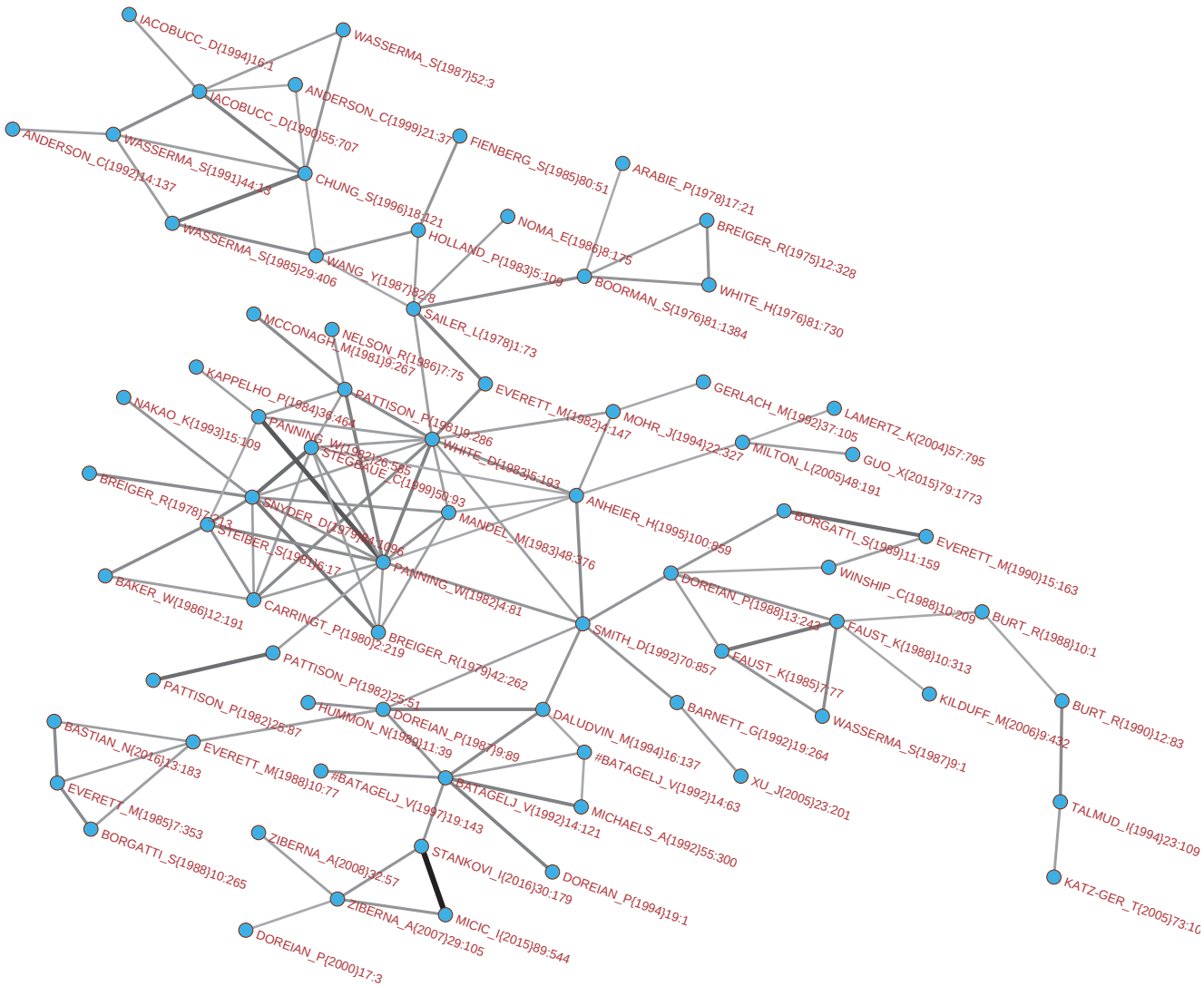
Network BMc (2016): for "block model*" or "network cluster*" ...;
$|W| = 5695$, $|A| = 13376$, $|J| = 1756$, $|K| = 10269$

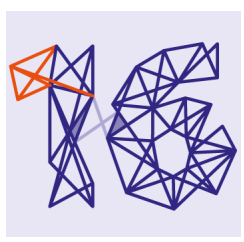




Bibliographic Coupling

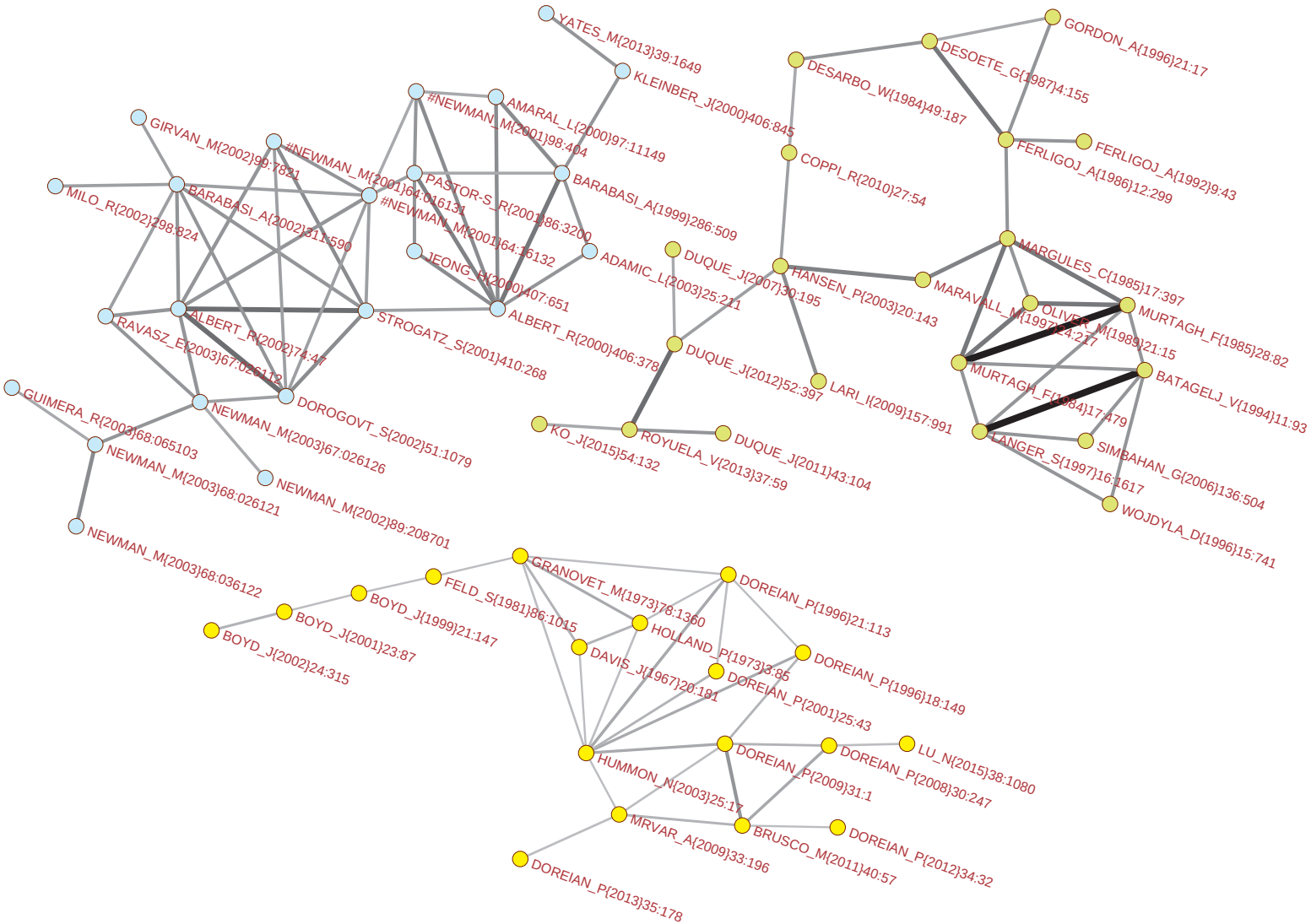
Jaccard island 4 (74)

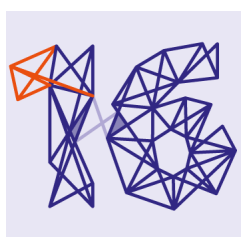




Bibliographic Coupling

Jaccard islands 12 (23), 11 (22), 1 (18)

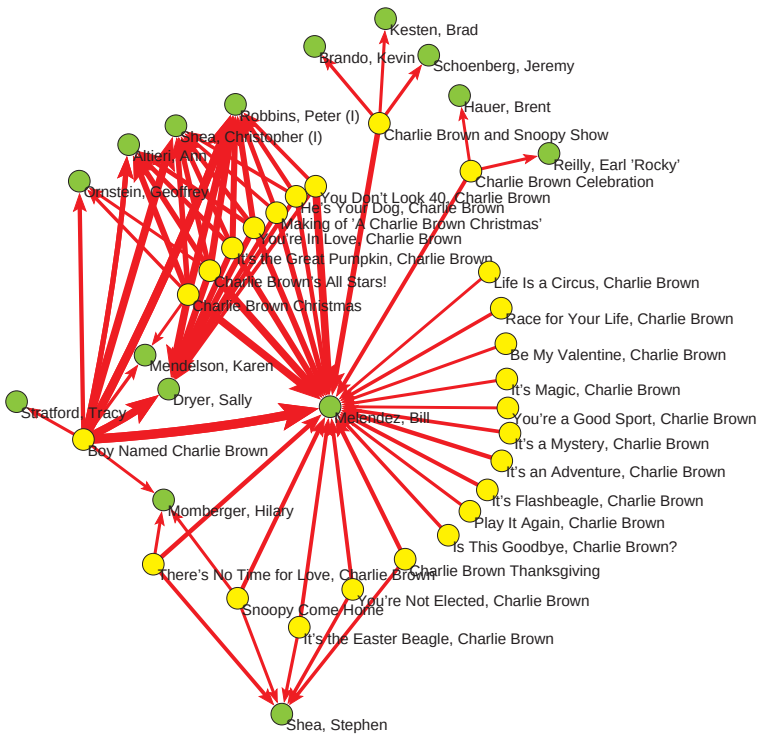


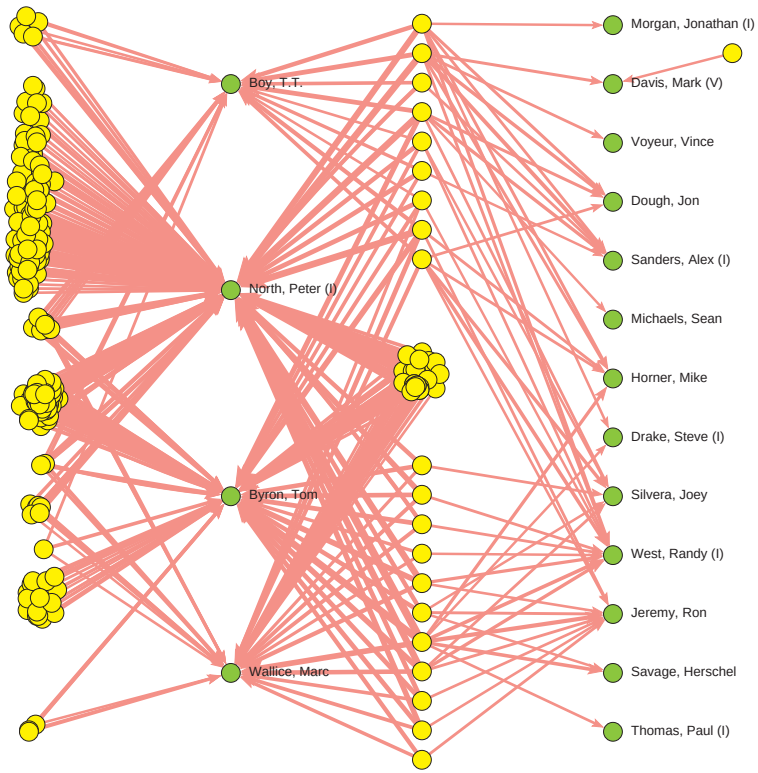


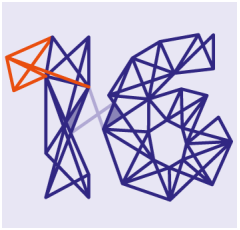
Example: Islands for w_4

Charlie Brown and Adult

IMDB: $n_1 = 428440,\ n_2 = 896308,\ m = 3792390.$





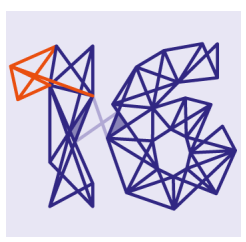


Clustering elements of a network

Usually we are clustering the nodes of a given network.

There are different approches:

- connectivity components from graph theory
- clustering with relational constraint
- blockmodeling

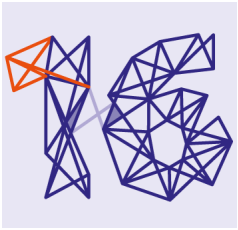


Graph theoretic approaches

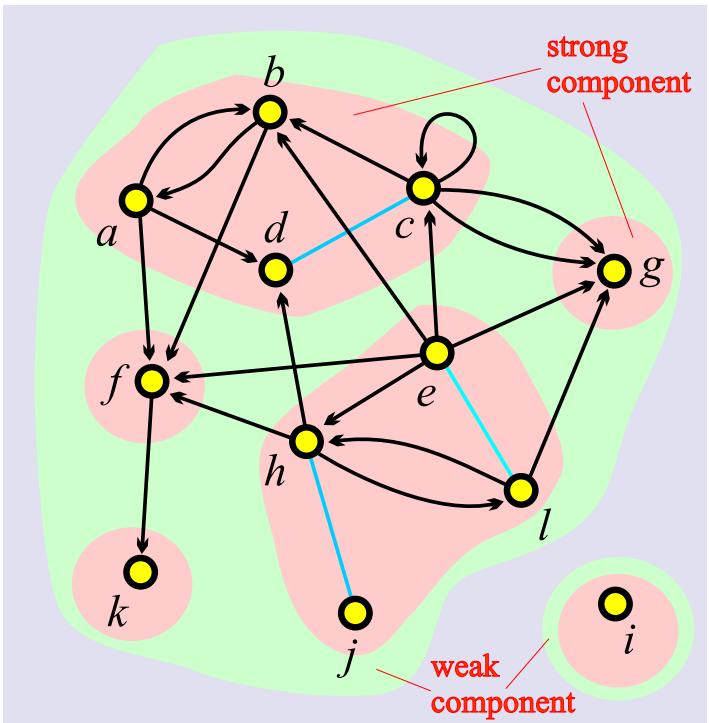
The basic decomposition of undirected graphs is to (weakly) connected components – partition of nodes (and links); and to (weakly) biconnected components – partition of links. For both, very efficient algorithms exist [Cormen et al.(2001)].

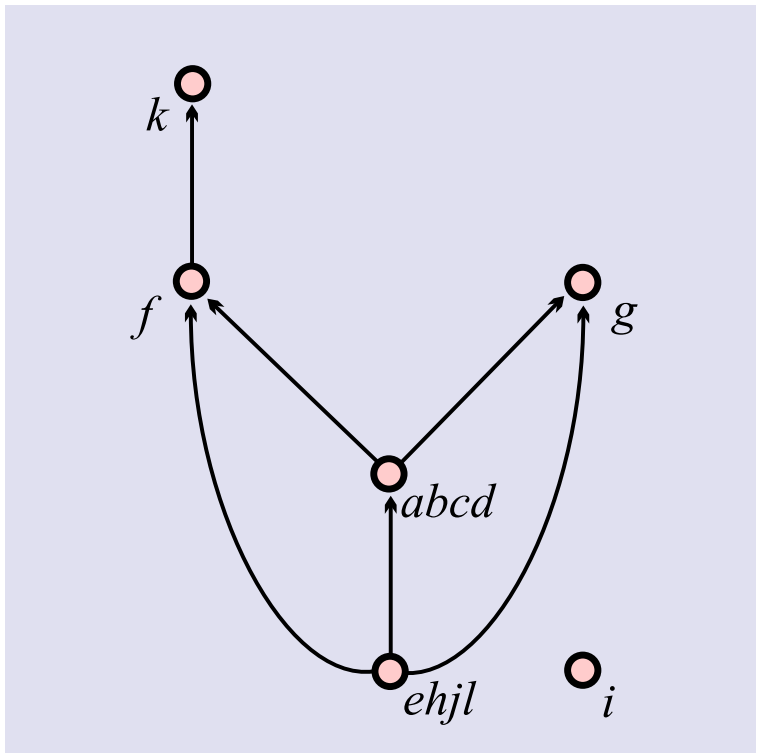
For directed graphs, the fundamental decomposition results can be found in [Cartwright and Harary(1956)]. If in a directed graph we shrink each strong component into a node the obtained reduced graph (*condensation*) is acyclic.

In the 1970s and 1980s, Matula studied different types of connectivities in graphs and the structures they induce [Matula(1977)]. In most cases the algorithms are too demanding to be used on larger graphs. A nice overview of connectivity algorithms can be found in [Esfahanian(2013)]. The graph partitioning problem has also several technical applications supported by special algorithms [Kernighan and Lin(1970)], [Karypis and Kumar(1998)], [Grygorash et al.(2006)].

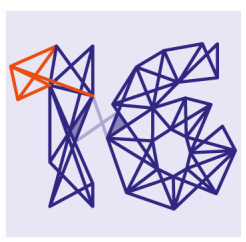


Condensation





If we shrink every strong component of a given graph into a node, delete all loops and identify parallel arcs the obtained *reduced* graph (*condensation*) is acyclic. For every acyclic graph an *ordering* / *level* function $i : \mathcal{V} \rightarrow \mathbb{N}$ exists s.t. $(u, v) \in \mathcal{A} \Rightarrow i(u) < i(v)$.



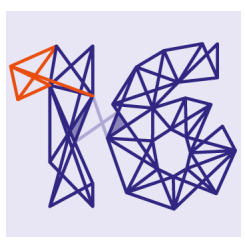
Clustering with relational constraint

We shall deal with the *clustering problem* (Φ, P):
Determine the clustering $\mathbf{C}^\star \in \Phi$ for which

$$P(\mathbf{C}^\star) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$

where $\mathbf{U}$ is a finite *set of units*; C is a *cluster*, $\emptyset \subset C \subseteq \mathbf{U}$; $\mathbf{C} = \{C_i\}$ is a *clustering*; Φ is a set of *feasible clusterings*; and $P : \Phi \to \mathbb{R}_0^+$ is a *criterion function*.

The criterion function $P(\mathbf{C})$ combines "partial/local errors" into a "total error". Usually it takes the form: $P(\mathbf{C}) = \sum_{C \in \mathbf{C}} p(C)$. The cluster-error $p(C)$ is usually expressed using a *dissimilarity* $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}_0^+$.



Agglomerative method for relational constraints

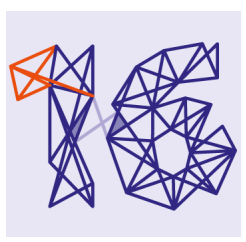
Suppose that the units are described by attribute data a: $\mathbf{U} \rightarrow [\mathbf{U}]$ and are related by a binary *relation* $R \subseteq \mathbf{U} \times \mathbf{U}$ that determine the *relational data* or *network* $(\mathbf{U}, R, a)$ [Batagelj and Ferligoj(2000)].

We want to cluster the units according to some (dis)similarity of their descriptions, but also considering the relation R which imposes *constraints* on the set of feasible clusterings [Ferligoj and Batagelj(1982)], [Ferligoj and Batagelj(1983)], usually in the following form:

$$\Phi(R) = \{\mathbf{C} \in P(\mathbf{U}) : \text{each cluster } C \in \mathbf{C} \text{ induces a subgraph } (C, R \cap C \times C) \text{ in the graph } (\mathbf{U}, R) \text{ of the required type of connectedness}\}$$

Example: regionalization problem – group given territorial units into regions such that units inside the region will be similar and form contiguous part of the territory.

We can define different types of sets of feasible clusterings for the same relation R.

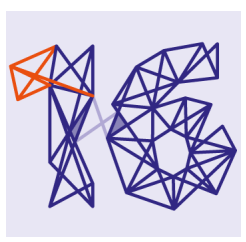


Types of connectedness

Some examples of *types of relational constraints*, $\Phi^i(R)$, are [Ferligoj and Batagelj(1983)]

clusterings	type of connectedness
$\Phi^1(R)$	weakly connected units
$\Phi^2(R)$	weakly connected units that contain at most one center
$\Phi^3(R)$	strongly connected units
$\Phi^4(R)$	clique
$\Phi^5(R)$	existence of a trail containing all the units of the cluster

In a directed graph a *trail* is a walk in which all arcs are distinct. The set $R(X) = \{Y : X\,R\,Y\}$ is a *set of successors* of unit $X \in \mathbf{U}$ and, for a cluster $C \subseteq \mathbf{U}$, $R(C) = \bigcup_{X \in C} R(X)$. A set of units, $L \subseteq C$ is a *center* of a cluster C in the clustering of type $\Phi^2(R)$ iff the subgraph induced by L is strongly connected and $R(L) \cap (C \setminus L) = \emptyset$.



Sets of feasible clusterings

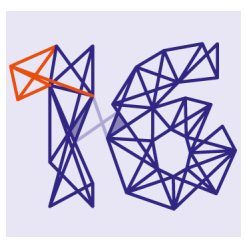
The sets of feasible clusterings $\Phi^i(R)$ are linked as follows: $\Phi^4(R) \subseteq \Phi^3(R) \subseteq \Phi^2(R) \subseteq \Phi^1(R)$ and $\Phi^4(R) \subseteq \Phi^5(R) \subseteq \Phi^2(R)$. If the relation R is symmetric, then $\Phi^3(R) = \Phi^1(R)$. If the relation R is an equivalence relation, then $\Phi^4(R) = \Phi^1(R)$.

The corresponding fusibility predicates are as follows:

$$\psi^1(C_1, C_2) \equiv \exists X \in C_1 \exists Y \in C_2 : (XRY \vee YRX)$$
$$\psi^2(C_1, C_2) \equiv (\exists X \in L_1 \exists Y \in C_2 : XRY) \vee (\exists X \in C_1 \exists Y \in L_2 : YRX)$$
$$\psi^3(C_1, C_2) \equiv (\exists X \in C_1 \exists Y \in C_2 : XRY) \wedge (\exists X \in C_1 \exists Y \in C_2 : YRX)$$
$$\psi^4(C_1, C_2) \equiv \forall X \in C_1 \forall Y \in C_2 : (XRY \wedge YRX)$$
$$\psi^5(C_1, C_2) \equiv (\exists X \in T_1 \exists Y \in I_2 : XRY) \vee (\exists X \in I_1 \exists Y \in T_2 : YRX)$$

where I denotes initial nodes in a cluster C and T denotes terminal nodes in a cluster C. For ψ^3 the property F5 fails.

We can use both hierarchical and local optimization methods for solving some types of problems with relational constraint [Ferligoj and Batagelj(1982), Ferligoj and Batagelj(1983), Batagelj et al.(2014)].



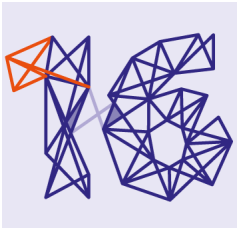
Hierarchical method

Here, we present only the hierarchical method:

1. $k := n$; $\mathbf{C}(k) := \{\{X\} : X \in \mathbf{U}\}$; $h_D(X) = 0, X \in \mathbf{U}$
2. **while** $\exists C_i, C_j \in \mathbf{C}(k)$: $(i \neq j \wedge \psi(C_i, C_j))$ **repeat**
2.1. $(C_p, C_q) := \text{argmin}\{D(C_i, C_j) : i \neq j \wedge \psi(C_i, C_j)\}$;
2.2. $C := C_p \cup C_q$; $k := k - 1$; $h_D(C) = D(C_p, C_q)$
2.3. $\mathbf{C}(k) := \mathbf{C}(k+1) \setminus \{C_p, C_q\} \cup \{C\}$;
2.4. determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$; and
2.5. adjust the relation R as required by the clustering type and
3. $m := k$

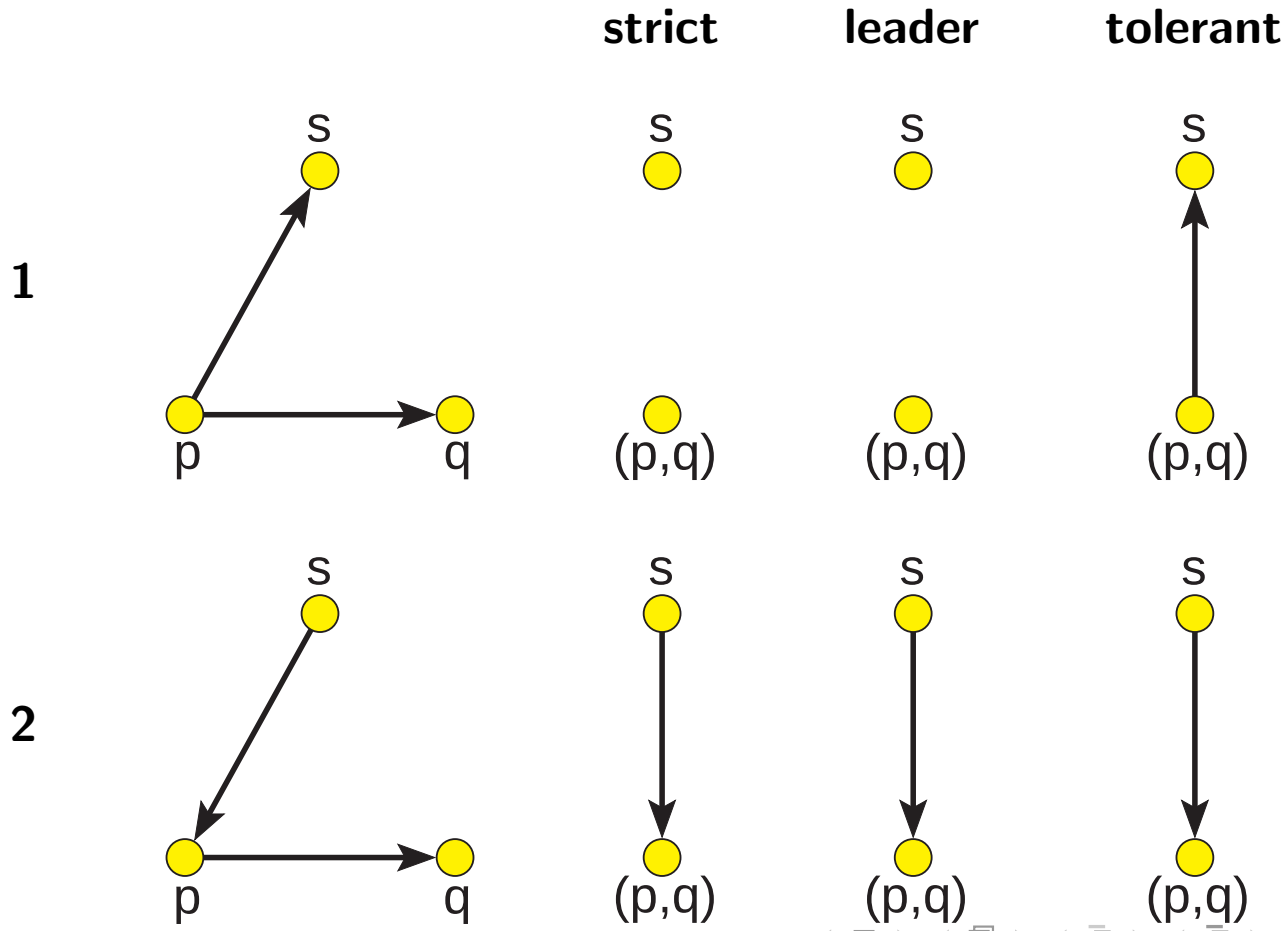
The *fusibility condition* $\psi(C_i, C_j)$ is equivalent to $C_i R C_j$ for tolerant, leader and strict method; and to $C_i R C_j \wedge C_j R C_i$ for two-way method.

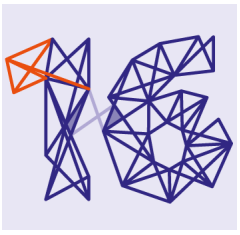
To get clustering procedures, it is necessary to further elaborate the questions how to adjust the relation after joining two clusters and how to update the dissimilarity $D(C, C_s)$.



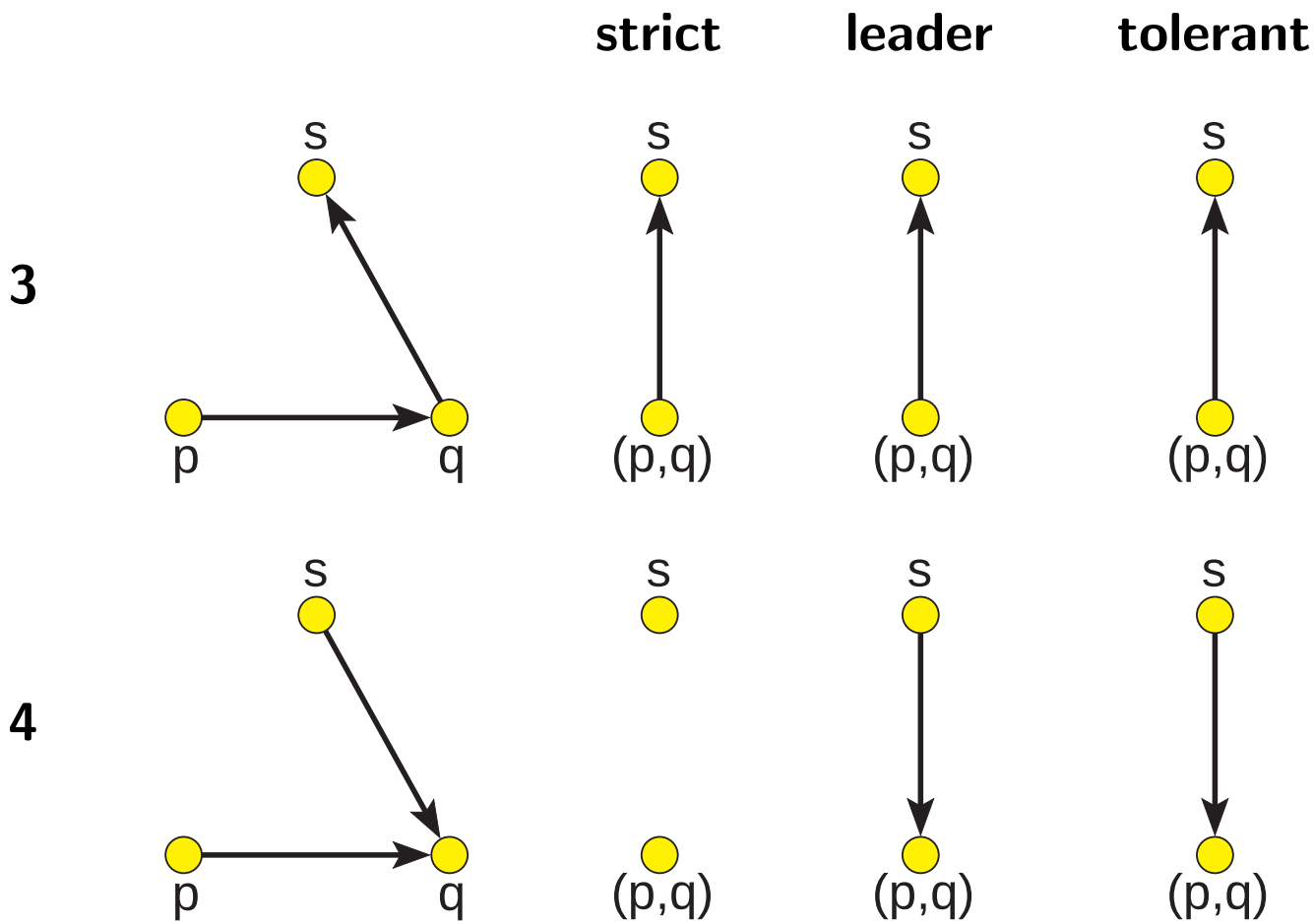
Types of relational constraints

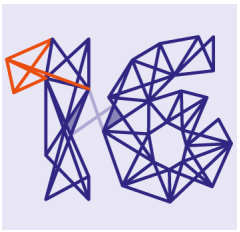
In figures four adjusting *strategies* are presented. They are compatible with the corresponding types of constraints: Φ^1 – tolerant, Φ^2 – leader, Φ^4 – strict, and Φ^5 – two.way.



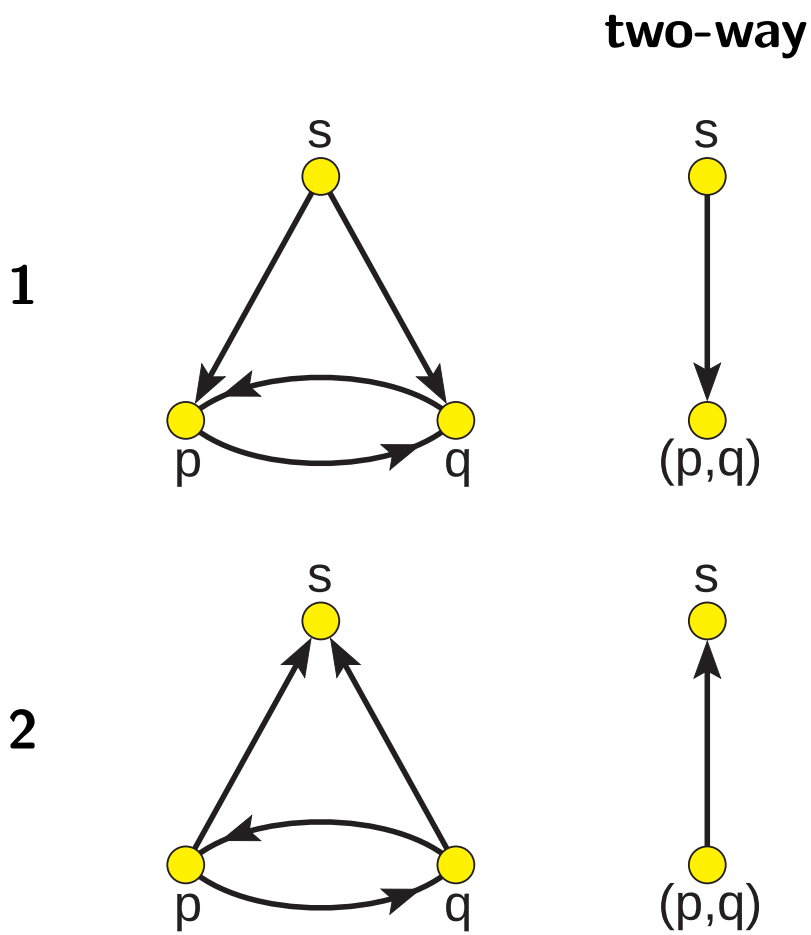


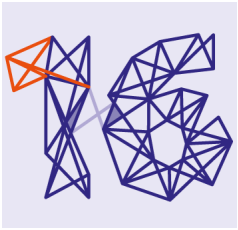
Types of relational constraints



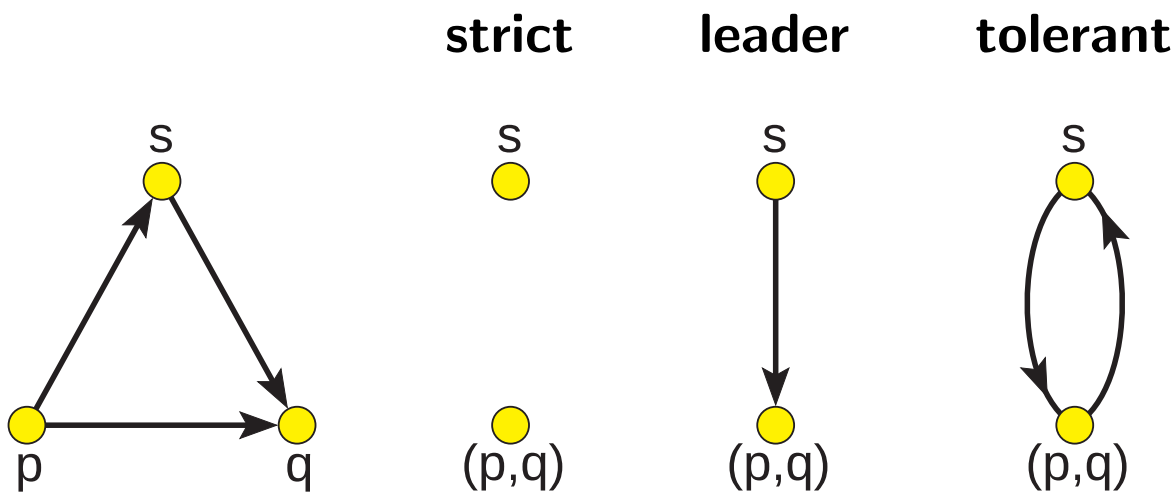


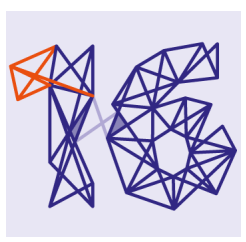
The two-way strategy





An example of application of strategies





Adjusting dissimilarity

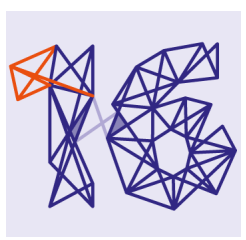
In the original approach (Ferligoj 1983), a complete dissimilarity matrix is needed. To obtain fast algorithms that can be applied to large data sets we propose *considering only the dissimilarities between linked units*. For large data sets, we assume that the relation R is *sparse*.

For step 2.4, "determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$" in the agglomerative procedure requires the adjustment of dissimilarities – computing the dissimilarities between a new cluster C and other remaining clusters. In the case of the relational constraints, we can limit the computation only to clusters that are related/linked to C.

This can be done efficiently in the following two ways:

A: we define a dissimilarity $D(S, T)$ between clusters S and T that allows quick updates (as in Lance-Williams formula).

B: to each cluster we assign a representative and can efficiently compute a representative of merged clusters along with a dissimilarity between clusters in terms of their representatives.



The first approach

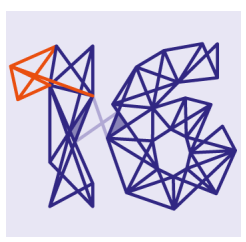
We will present only he first approach. Let $(\mathbf{U}, R)$, $R \subseteq \mathbf{U} \times \mathbf{U}$ be a graph and $\emptyset \subset S, T \subset \mathbf{U}$ and $S \cap T = \emptyset$. We call a *block* of relation R for S and T its part $R(S, T) = R \cap S \times T$. The *symmetric closure* of relation R we denote with $\hat{R} = R \cup R^{-1}$. It holds: $\hat{R}(S, T) = \hat{R}(T, S)$.
For all dissimilarities between clusters $D(S, T)$ we set:

$$D(\{s\}, \{t\}) = \begin{cases} d(s,t) & s \,\hat{R}\, t \\ \infty & \text{otherwise} \end{cases}$$

where d is a selected dissimilarity between units.
Minimum

$$D_{\min}(S, T) = \min_{(s,t) \in \hat{R}(S,T)} d(s, t)$$

$$D_{\min}(S, T_1 \cup T_2) = \min(D_{\min}(S, T_1), D_{\min}(S, T_2))$$



The first approach

Maximum

$$D_{\max}(S,T) = \max_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$D_{\max}(S, T_1 \cup T_2) = \max(D_{\max}(S, T_1), D_{\max}(S, T_2))$$

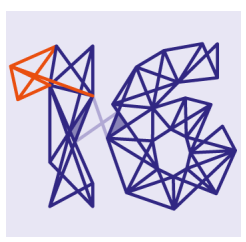
Average

$w : V \to \mathbb{R}$ – is a weight on units; for example $w(v) = 1$, for all $v \in \mathbf{U}$.

$$D_{\mathrm{a}}(S,T) = \frac{1}{w(\hat{R}(S,T))} \sum_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$w(\hat{R}(S, T_1 \cup T_2)) = w(\hat{R}(S, T_1)) + w(\hat{R}(S, T_2))$$

$$D_{\mathrm{a}}(S, T_1 \cup T_2) = \frac{w(\hat{R}(S, T_1))}{w(\hat{R}(S, T_1 \cup T_2))} D_{\mathrm{a}}(S, T_1) + \frac{w(\hat{R}(S, T_2))}{w(\hat{R}(S, T_1 \cup T_2))} D_{\mathrm{a}}(S, T_2)$$



Reducibility

The dissimilarity D has the *reducibility* property (Bruynooghe, 1977) iff for all C_p, C_q and C_s

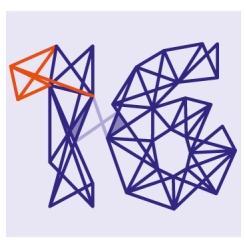
$$D(C_p, C_q) \leq \min(D(C_p, C_s), D(C_q, C_s)) \Rightarrow$$

$$\min(D(C_p, C_s), d(C_q, C_s)) \leq D(C_p \cup C_q, C_s)$$

Theorem: If a dissimilarity D has the reducibility property then h_D is a level function.

Theorem: Dissimilarities D_{min}, D_{max} and D_a have the reducibility property.

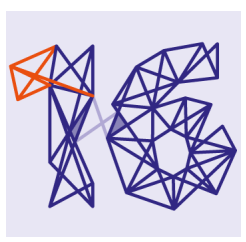
All three dissimilarities have the reducibility property. In this case, also the *nearest neighbor network* for a given network is preserved after joining the nearest clusters. This allows us to develop a very fast agglomerative hierarchical clustering procedure [Murtagh(1985)] and [Batagelj et al.(2014)] (Subsection 9.3.5). It is available in the program `Pajek`. The same approach can be extended also to clustering of links of network [Bodlaj and Batagelj(2015)] by transforming a given network into its line-graph in which the original links become new nodes.



Examples

To illustrate the hierarchical clustering with relational constraints, we use three examples:

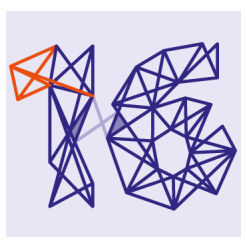
- Clustering the US states according to the selected variables into geographically contiguous clusters.
- Clustering of 3000 US counties.
- Clustering the authors from the network clustering literature (see Chapter 2) according to their citations into clusters with a single leaders group.



Example: US 2016 data

From the site https://datausa.io/profile/geo/united-states/ we obtained the data about US states in 2016 for the following variables: `crime` – homicide deaths, `violent` – violent crimes, `smoking` – adult smoking prevalence, `drinking` – excessive drinking prevalence, `diabetes` – diabetes prevalence, `opioid` – opioid overdose death rate, and `income` – median household income.

In his book *The Stanford GraphBase* [Knuth(1993)] Knuth provided a description of neighboring relation for the contiguous part of USA `contiguous-usa.dat` (without Alaska and Hawai). Because of missing data we removed also Washington DC.

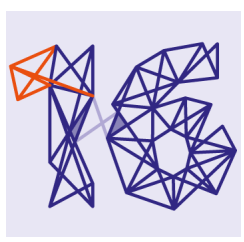


US 2016 data

We first applied the Ward's hierarchical clustering method using the squared Euclidean dissimilarity between units with standardized variables. On the basis of the corresponding dendrogram (see the left top part of the figure in next slide, we considered a clustering into 5 clusters:

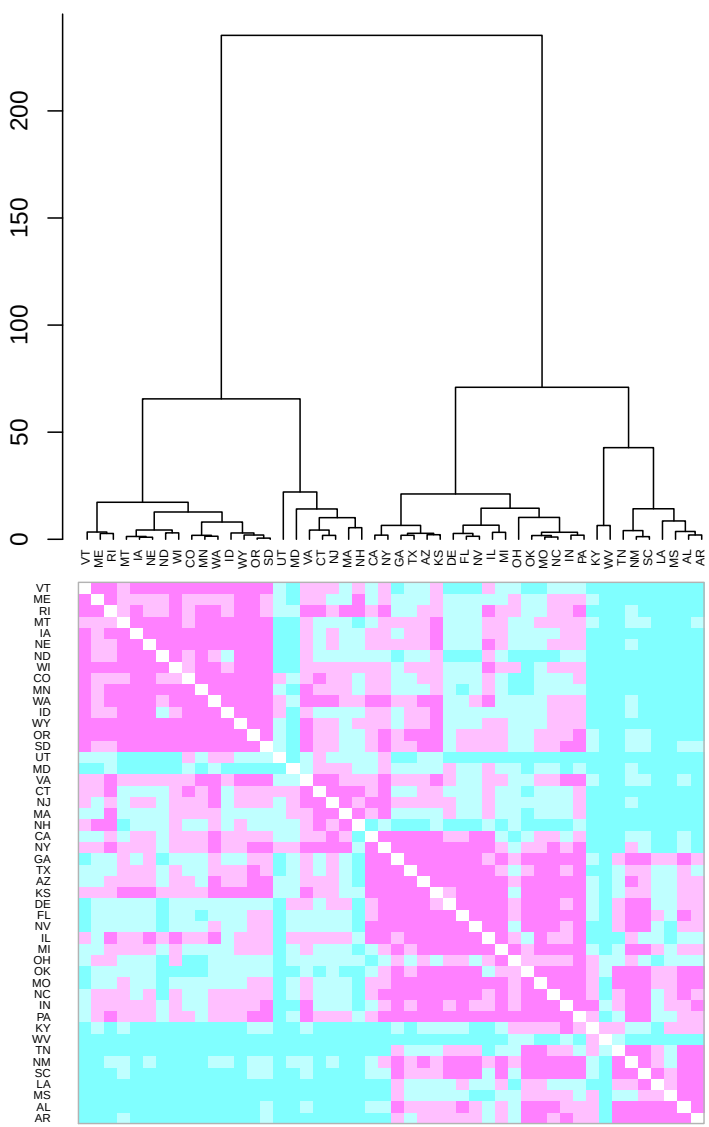
$$C_1 = \{AL, AR, LA, MS, NM, TN, SC\},$$
$$C_2 = \{AZ, CA, DE, FL, GA, IL, IN, KS, MI, MO, NC, NV, NY, OH, OK, PA, TX\},$$
$$C_3 = \{CO, IA, ID, ME, MN, MT, ND, NE, OR, SD, WY, RI, WI, WA, VT\},$$
$$C_4 = \{CT, MA, MD, NH, NJ, UT, VA\},$$
$$C_5 = \{KY, WV\}.$$

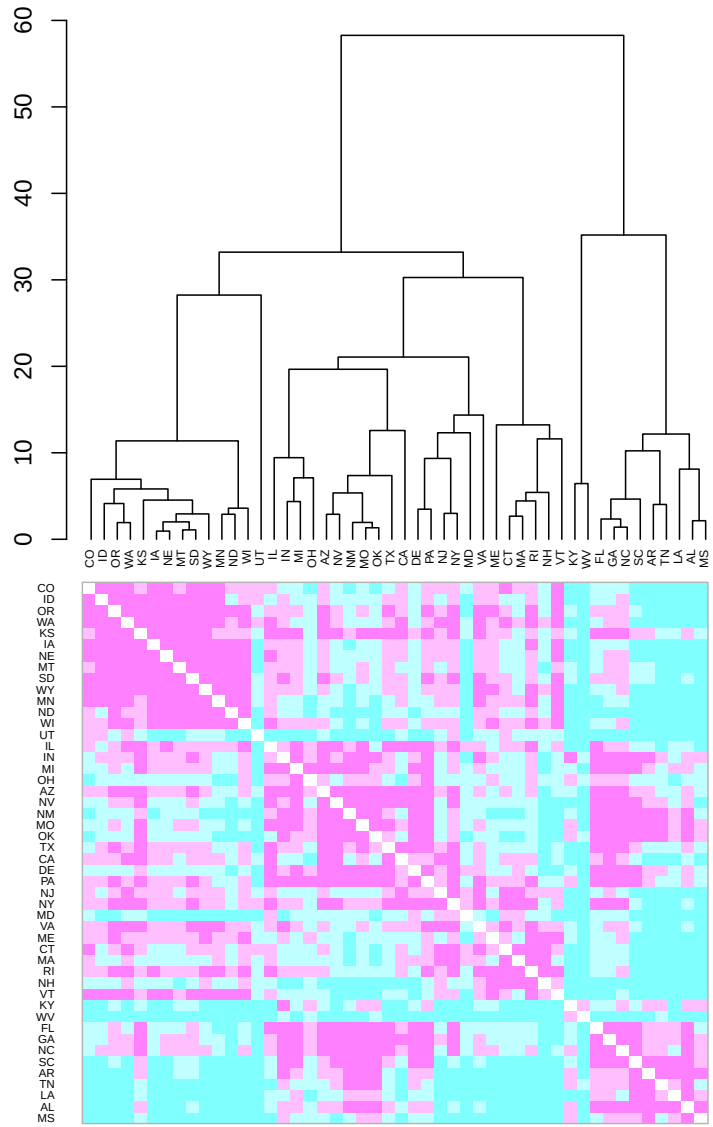
The bottom left part of the figure shows the dissimilarity matrix reordered according to the obtained clustering.

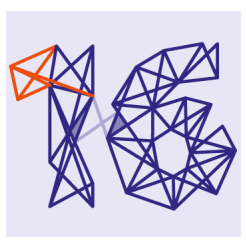


US 2016 data

Ward clustering (left) and Maximum/Tolerant clustering (right)





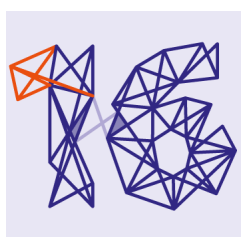


US 2016 data

Using hierarchical clustering with relational constraints with the Maximum/Tolerant strategy, we get a clustering that considers a given dissimilarity among units and produces clusters that form contiguous regions. On the basis of the dendrogram in the right top part of the previous figure, we considered a clustering into 6 clusters:

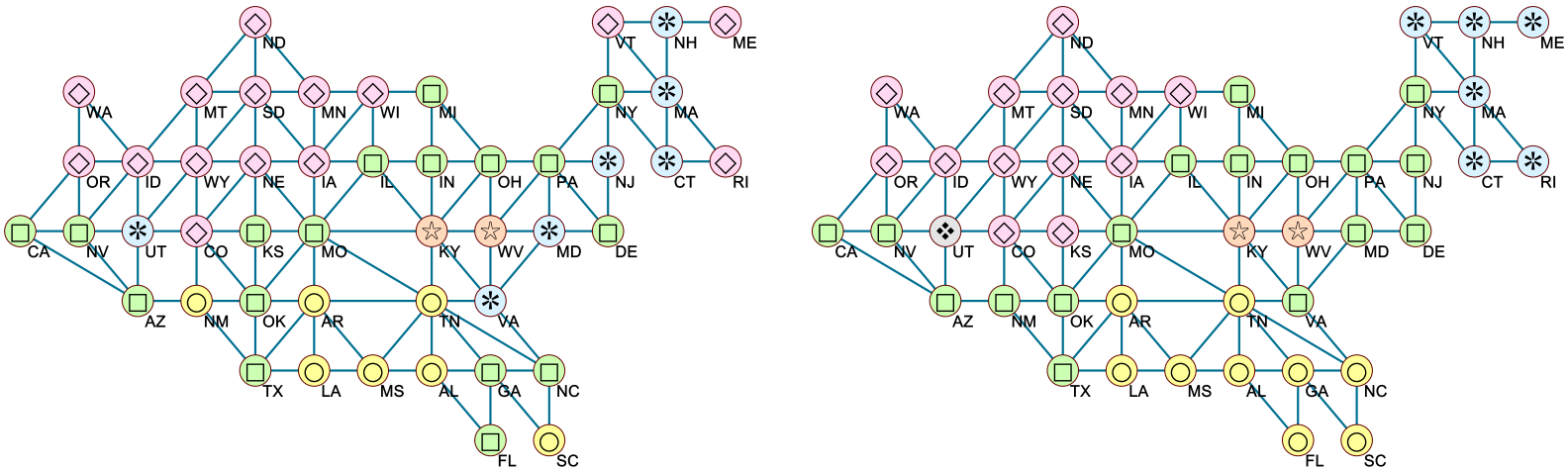
$C_1 = \{AL, AR, FL, GA, LA, MS, NC, TN, SC\}$,
$C_2 = \{AZ, CA, DE, IL, IN, MD, MI, MO, NJ, NM, NV, NY, OH, OK, PA, VA, TX\}$,
$C_3 = \{CO, IA, ID, KS, MN, MT, ND, NE, OR, SD, WY, WI, WA\}$,
$C_4 = \{CT, MA, ME, NH, RI, VT\}$,
$C_5 = \{KY, WV\}$,
$C_6 = \{UT\}$.

The clusters of the obtained clustering/partition induce connected subnetworks, as expected. See the right network in the next figure.

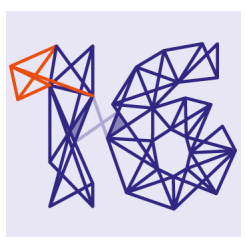


US 2016 data

Ward clustering (left) and Maximum/Tolerant clustering (right)

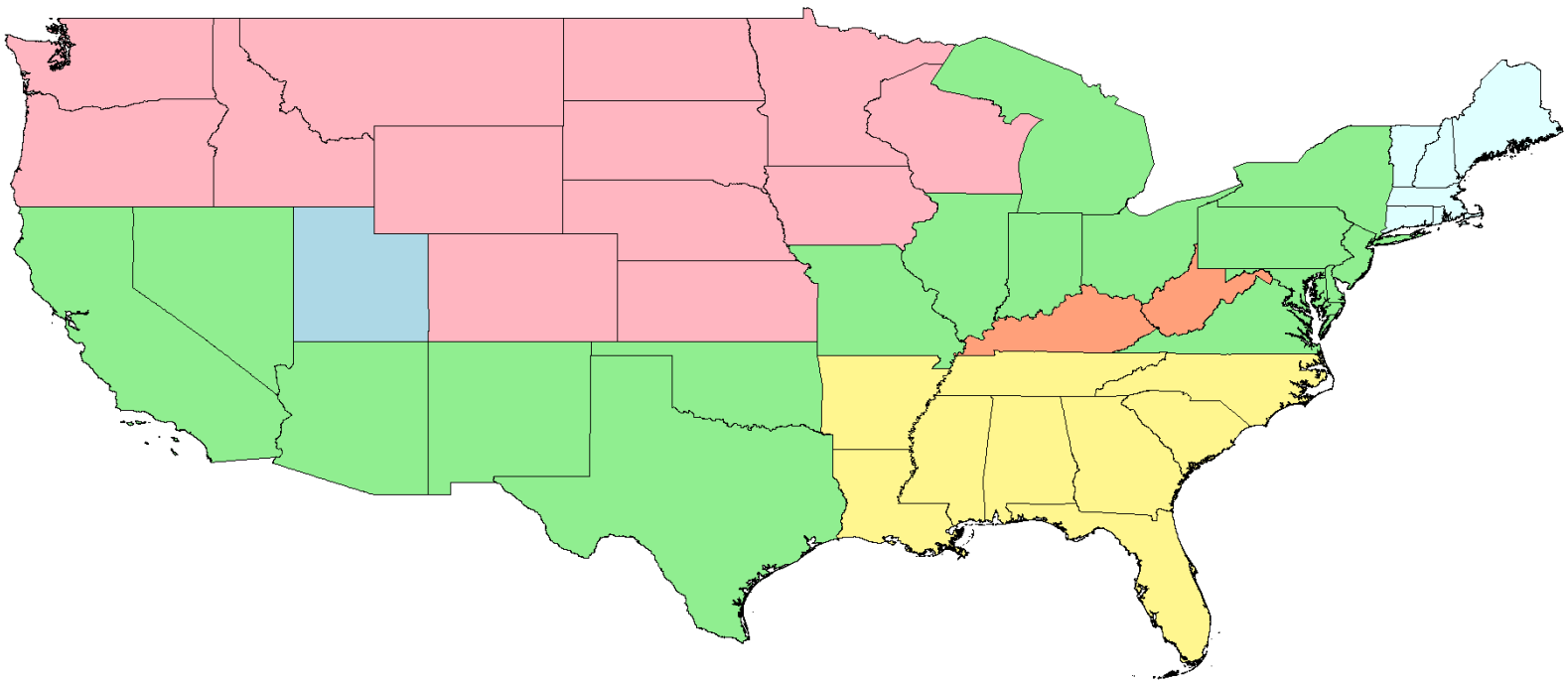


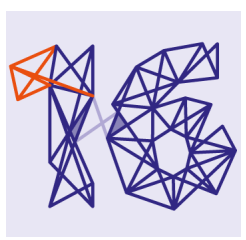
In the left part of figure the obtained clustering/partition is represented with node colors on the network of neighboring US states. It is clear that the subnetworks induced by clusters are not all connected (forming contiguous regions). For example, the subnetwork induced by C_4 has 4 components $\{CT, MA, NH\}$, $\{NJ\}$, $\{MD, VA\}$ and $\{UT\}$. The subnetworks in the right part of figure are all connected.



US 2016 data

Maximum/Tolerant partition on the map



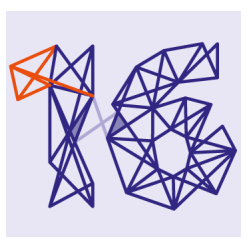


US 2016 data

Averages for Ward's clustering

To interpret the obtained clusters we produced a table with averages of each variable over each cluster for raw and standardized units. The interpretation is left to the reader.

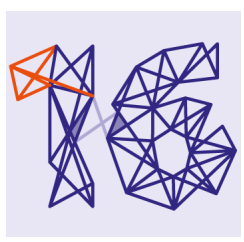
	crime	*violent*	*smoking*	*drinking*	*diabetes*	*opioid*	*income*
C_1	8.7857	496.45	0.2251	0.1447	0.1173	10.857	44631
C_2	5.9118	427.96	0.1826	0.1714	0.1048	13.853	53535
C_3	2.6333	239.99	0.1755	0.2023	0.0847	10.767	55908
C_4	3.8000	300.99	0.1521	0.1699	0.0903	23.657	69947
C_5	4.9000	273.02	0.2645	0.1195	0.1210	33.500	43727
all	4.9563	354.23	0.1856	0.1748	0.0989	14.700	54963
C_1	**1.5723**	**1.0924**	1.1363	−0.9927	1.2826	−0.4229	−1.1990
C_2	0.3923	0.5663	−0.0843	−0.1123	0.4134	−0.0932	−0.1657
C_3	**−0.9537**	**−0.8776**	−0.2887	**0.9094**	**−0.9924**	**−0.4328**	0.1097
C_4	−0.4747	−0.4090	**−0.9605**	−0.1617	−0.6005	0.9856	**1.7389**
C_5	−0.0231	−0.6239	**2.2668**	**−1.8260**	**1.5416**	**2.0687**	**−1.3039**



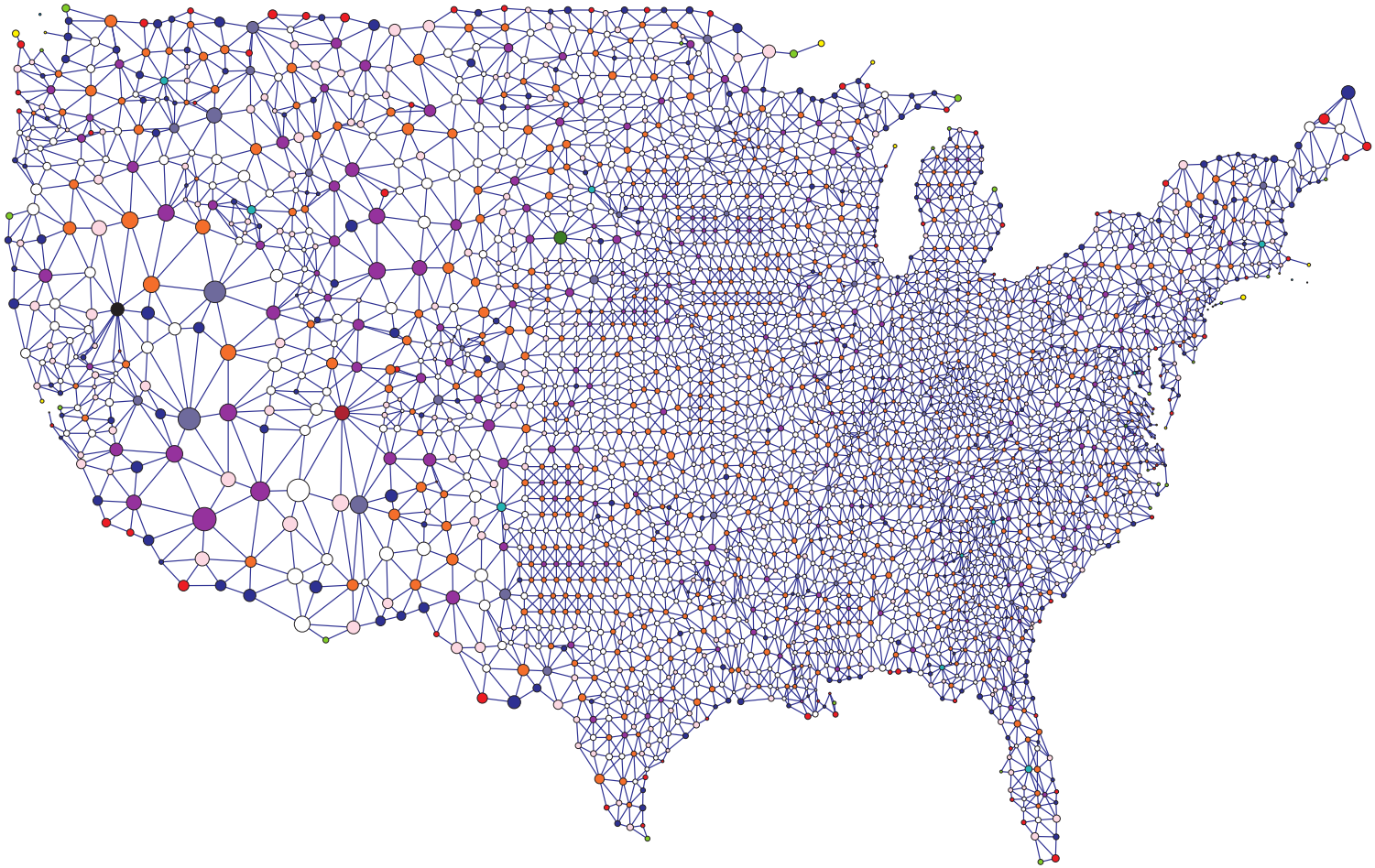
US 2016 data

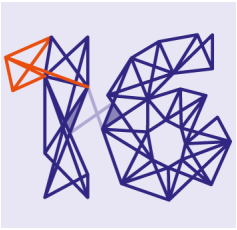
Averages for Maximum/Tolerant clustering

	crime	*violent*	*smoking*	*drinking*	*diabetes*	*opioid*	*income*
C_1	8.1667	462.00	0.2140	0.1488	0.1160	10.788	46104
C_2	5.9701	425.91	0.1804	0.1719	0.1032	15.794	57054
C_3	2.8385	265.25	0.1765	0.2005	0.0852	7.408	55913
C_4	2.3833	234.31	0.1660	0.1932	0.0880	26.717	62751
C_5	4.9000	273.02	0.2645	0.1195	0.1210	33.500	43727
C_6	1.9000	204.72	0.0970	0.1210	0.0710	16.400	62518
all	4.9563	354.23	0.1856	0.1748	0.0989	14.700	54963
C_1	**1.3181**	**0.8278**	0.8162	−0.8584	1.1929	−0.4304	−1.0281
C_2	0.4165	0.5506	−0.1502	−0.0928	0.3026	0.1204	0.2427
C_3	−0.8695	−0.6836	−0.2620	**0.8523**	−0.9584	**−0.8024**	0.1103
C_4	−1.0564	−0.9212	−0.5625	0.6087	−0.7599	1.3223	**0.9039**
C_5	−0.0231	−0.6239	**2.2668**	**−1.8260**	**1.5416**	**2.0687**	**−1.3039**
C_6	**−1.2548**	**−1.1485**	**−2.5445**	−1.7764	**−1.9456**	0.1871	0.8767

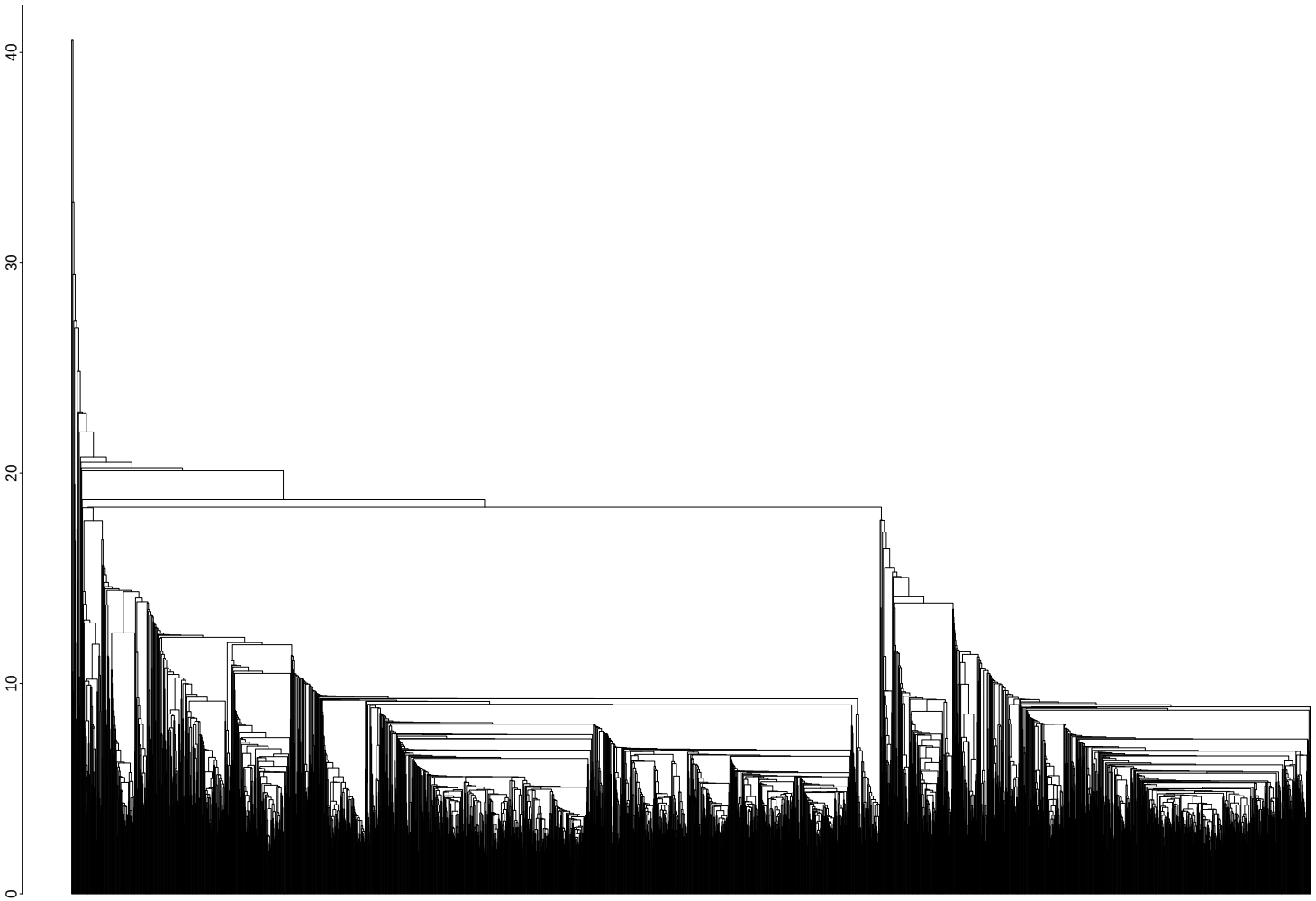


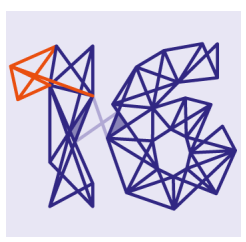
Example: US counties neighboring relation



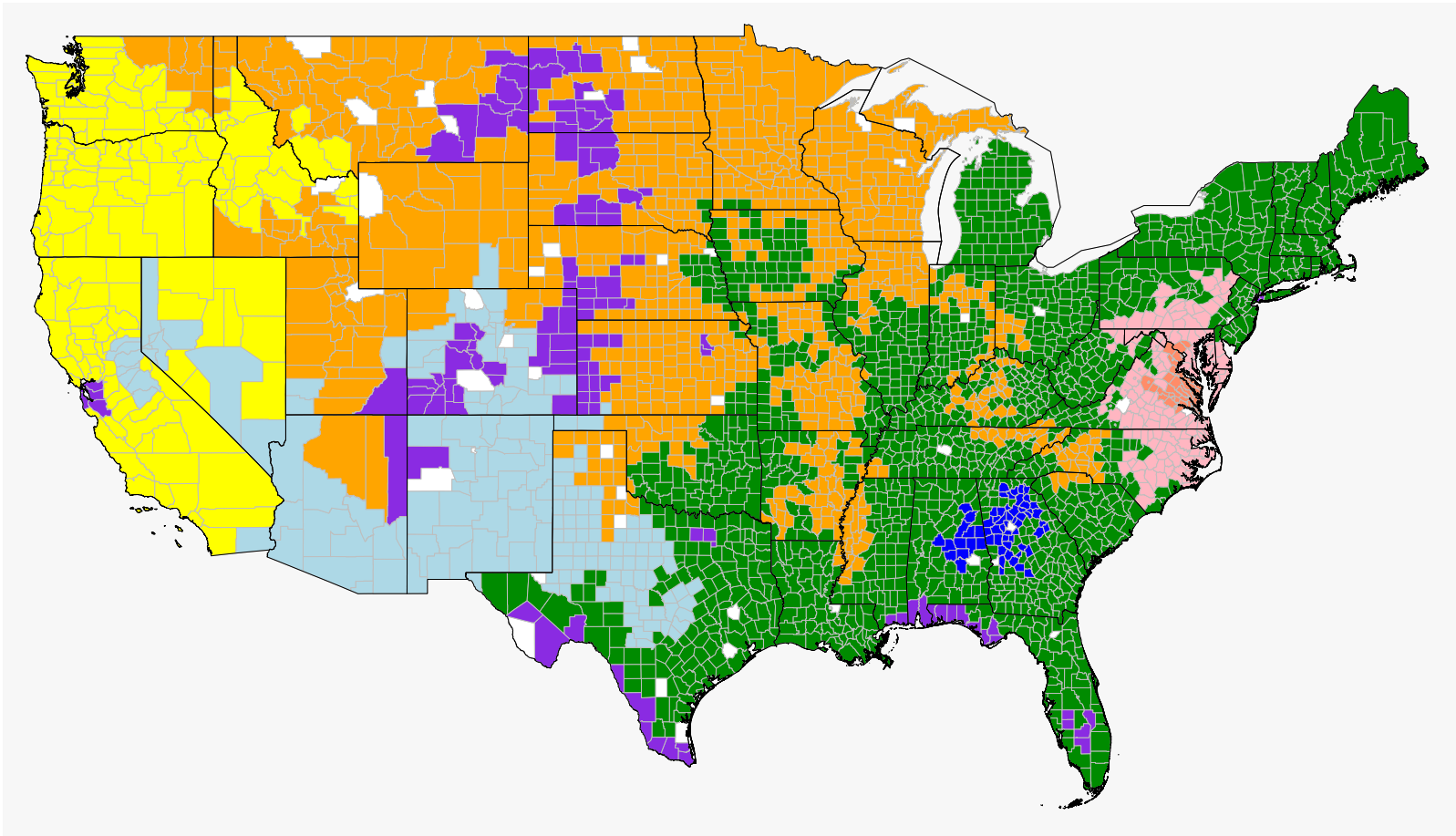


Example: US counties

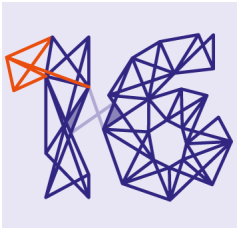




Example: US partition in 8 clusters – regions



We obtained many clusters. In the picture, we preserved only the largest eight regions with at least 20 counties. Smaller regions are colored with violet and the outliers with white.

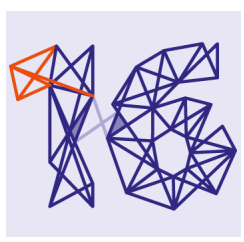


Example: Citations among authors from the network clustering literature

We consider the bibliometric data on the network clustering literature. We analyze the normalized network of citations among authors **nAcite** $= n(\mathbf{WAc})^T * n(\mathbf{CiteC}) * n(\mathbf{WAc})$. Every work has 1 point. They are distributed on arcs of the derived network. The weight **nAcite**$[u, v]$ of the arc (u, v) is equal to the fractional share of works co-authored by u that are citing a work co-authored by v.

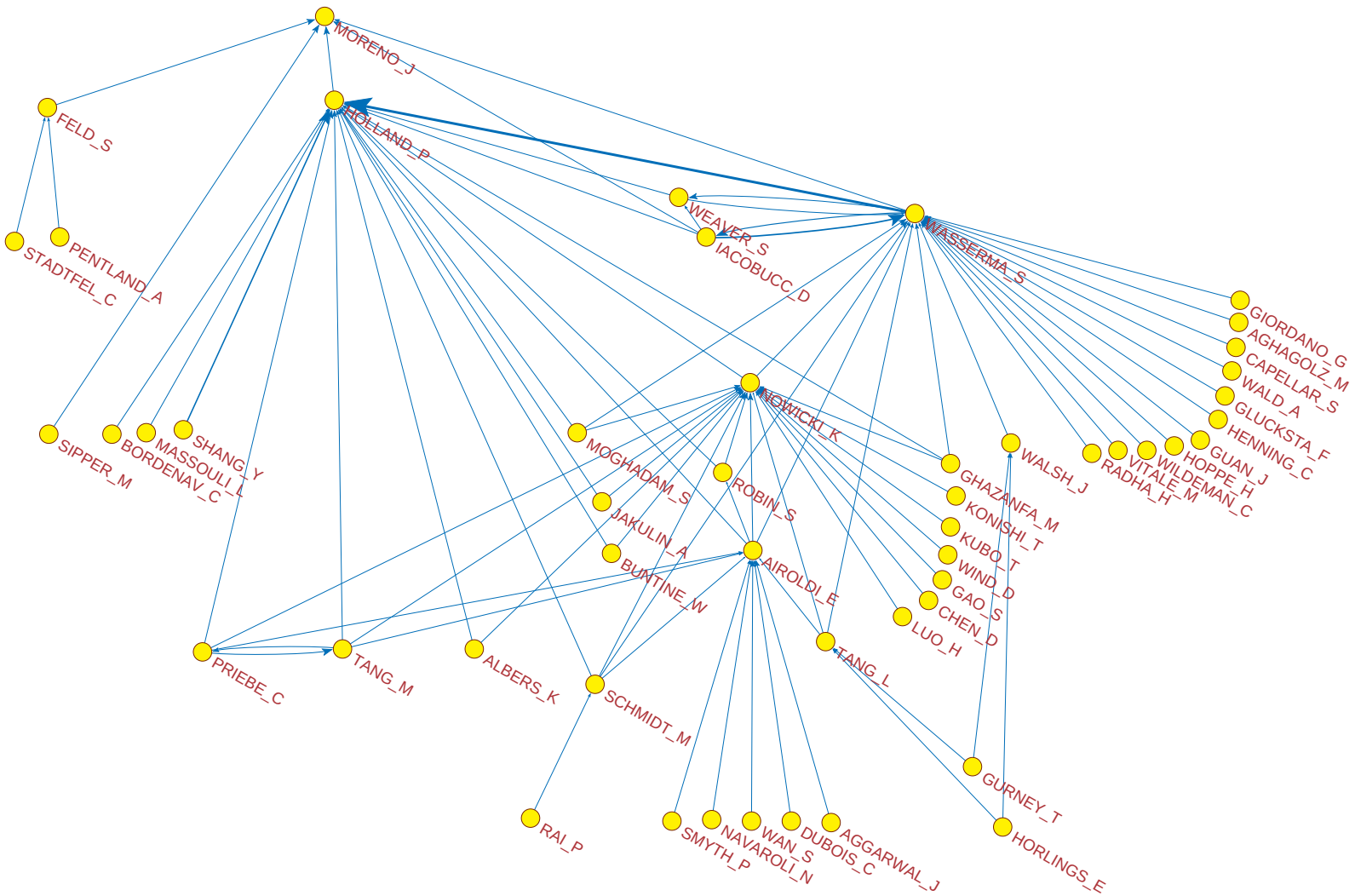
In this example, we identified clusters such that the corresponding induced subnetworks are connected and contain a single center – type Φ^2. The **nAcite** weights are similarities, $s \in [\infty, 0]$. To convert them to distances d, different transformations can be used, including: $d = \frac{s_{max}}{s} - 1 \in [0, \infty]$ or $d = 1 - \frac{s}{s_{max}} \in [0, 1]$. We selected the second option with $s_{max} = 2.52$.

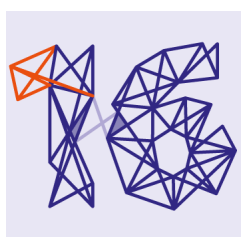
On the obtained network, we applied, in Pajek, the hierarchical clustering with relational constraints procedure with the Maximum/Leader strategy and determined the partition of units into clusters of size at most 50. There are 257 such clusters. To reduce their number, we decided to consider only clusters with at least 20 units. There are 57 such clusters. Most of the subnetworks of clusters for the Leader strategy have almost acyclic structure. This has to be considered also in their visualization.



Citations in network clustering literature

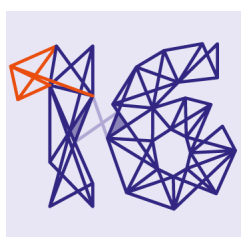
Wasserman's subnetwork





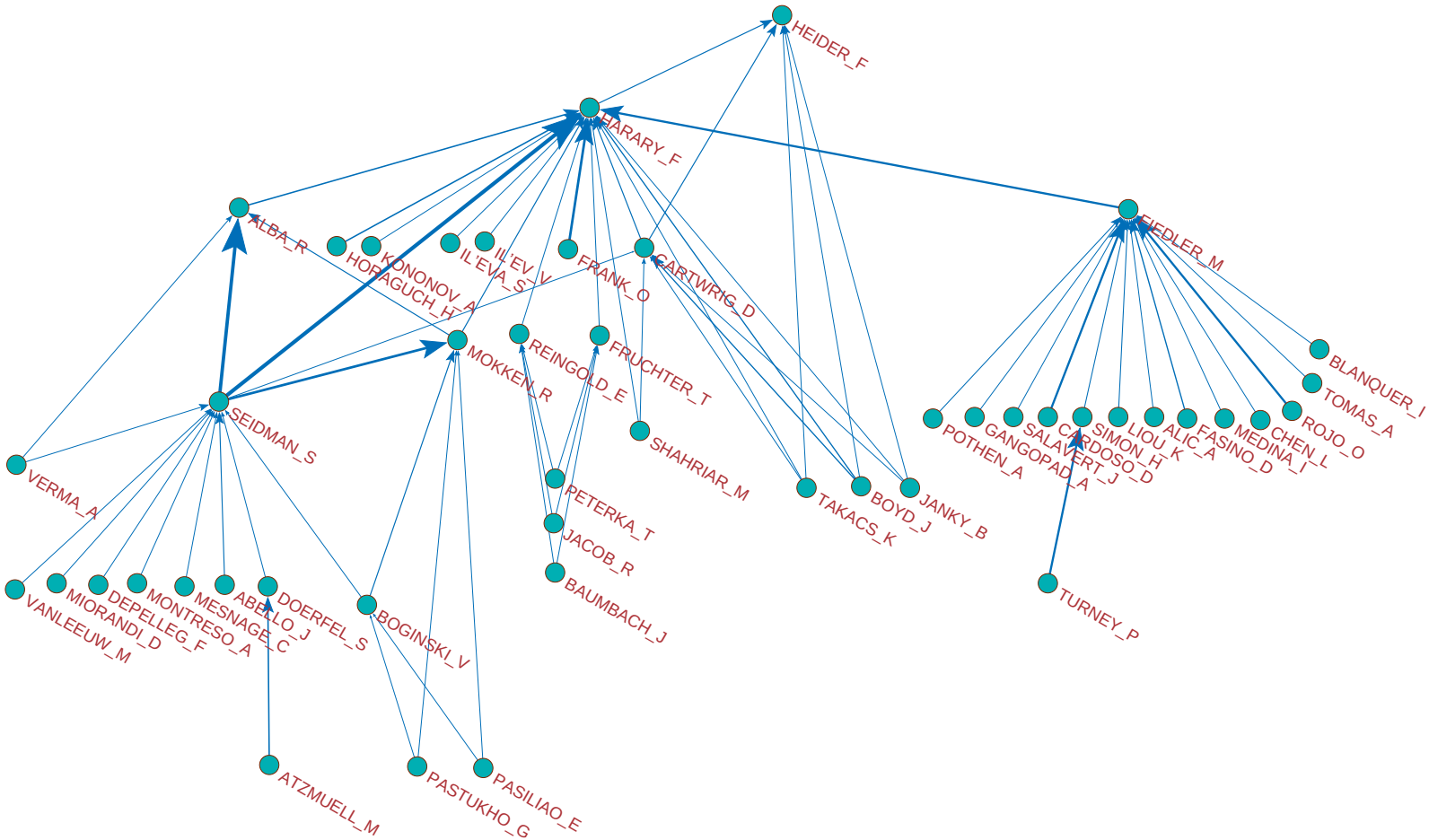
Citations in network clustering literature

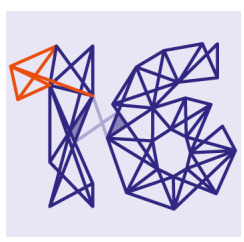
Ward's subnetwork



Citations in network clustering literature

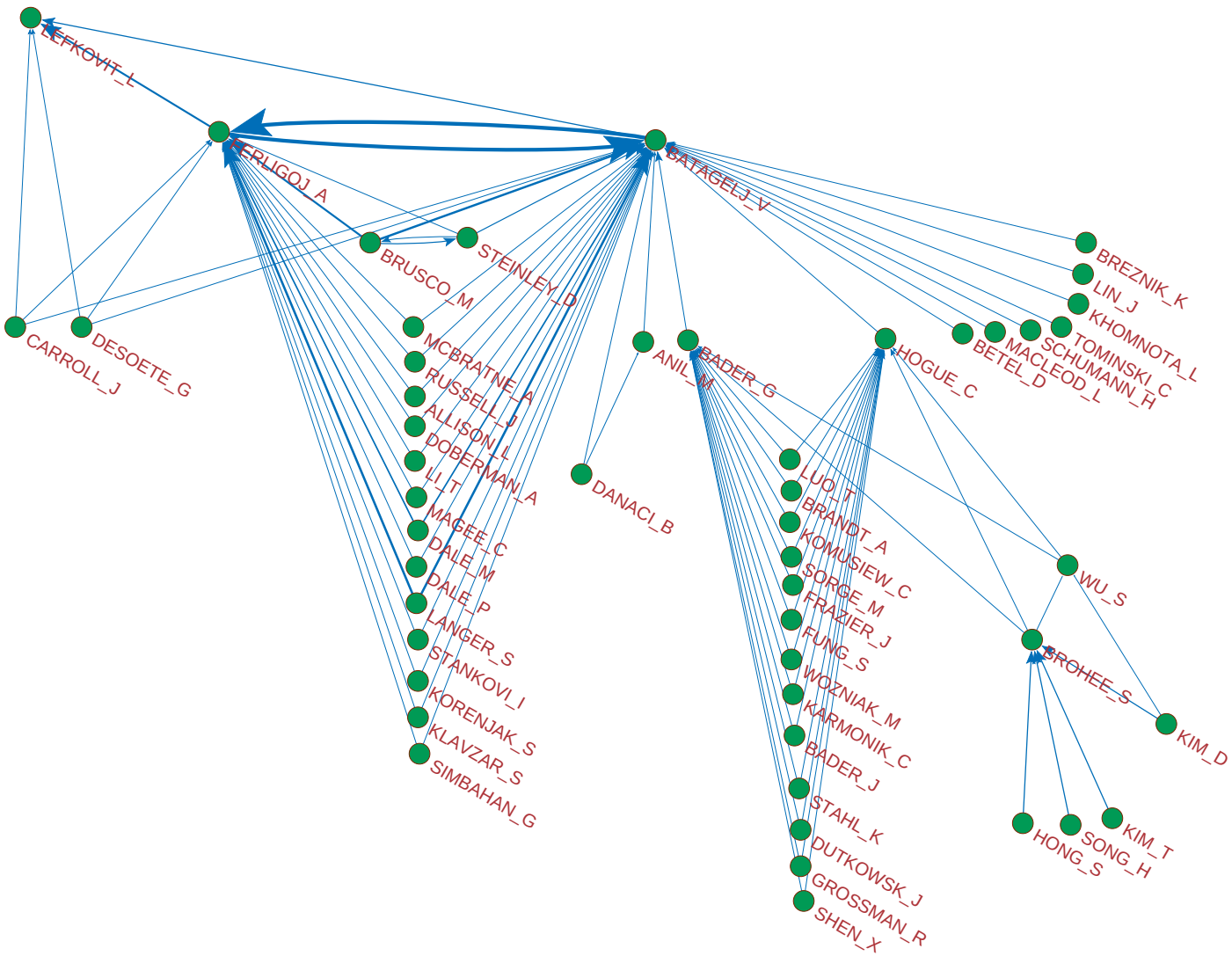
Harary's subnetwork

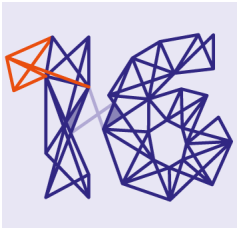




Citations in network clustering literature

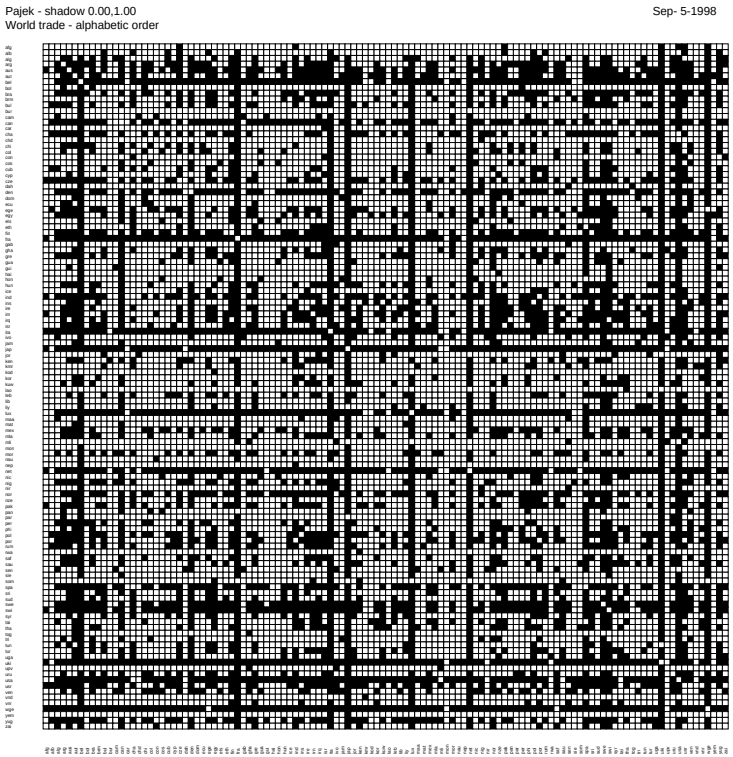
Batagelj + Ferligoj's subnetwork

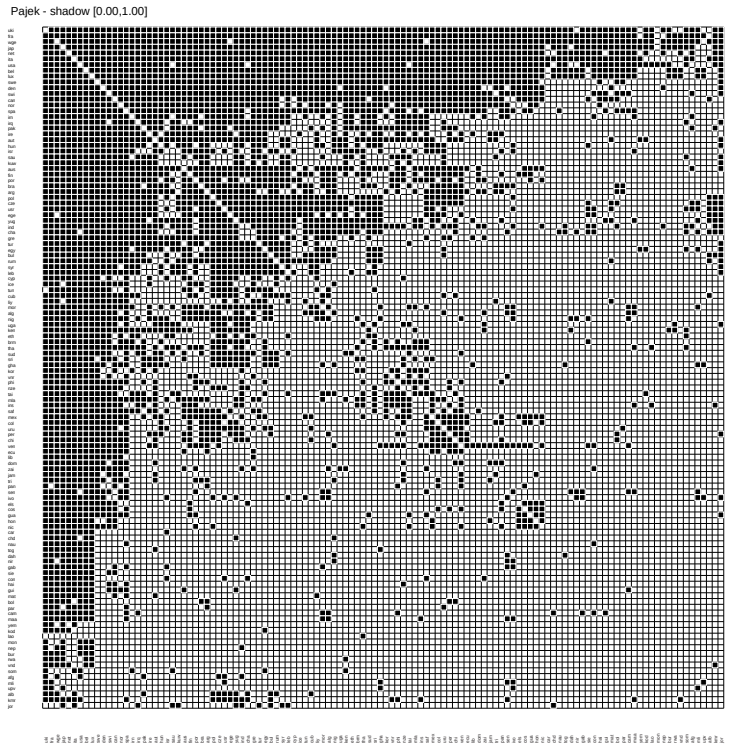




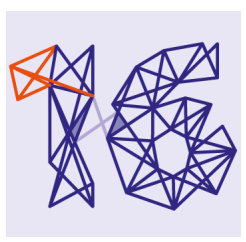
Matrix rearrangement view on blockmodeling

Snyder & Kick's World trade network / $n = 118$, $m = 514$



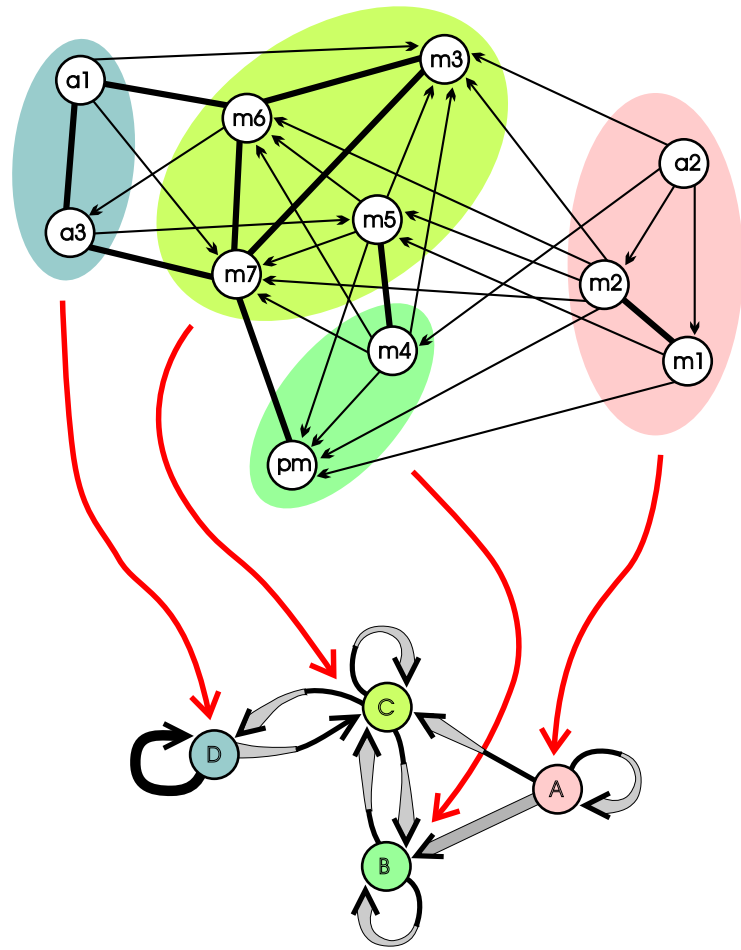


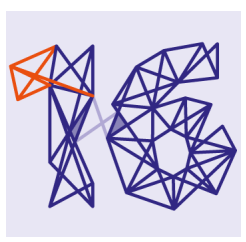
Alphabetic order of countries (left) and rearrangement (right)



Blockmodeling as a clustering problem

The goal of *blockmodeling* is to reduce a large, potentially incoherent network to a smaller comprehensible structure that can be interpreted more readily. Blockmodeling, as an empirical procedure, is based on the idea that units in a network can be grouped according to the extent to which they are equivalent, according to some *meaningful* definition of equivalence.



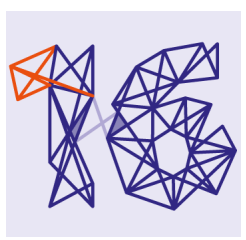


Cluster, clustering, blocks

One of the main procedural goals of blockmodeling is to identify, in a given network $\mathcal{N} = (\mathbf{U}, R)$, $R \subseteq \mathbf{U} \times \mathbf{U}$, *clusters* (classes) of units that share structural characteristics defined in terms of R. The units within a cluster have the same or similar connection patterns to other units. They form a *clustering* $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ which is a *partition* of the set $\mathbf{U}$. Each partition determines an equivalence relation (and vice versa). Let us denote by $\sim$ the relation determined by partition $\mathbf{C}$.
A clustering $\mathbf{C}$ partitions also the relation R into *blocks*

$$R(C_i, C_j) = R \cap C_i \times C_j$$

Each such block consists of units belonging to clusters C_i and C_j and all arcs leading from cluster C_i to cluster C_j. If $i = j$, a block $R(C_i, C_i)$ is called a *diagonal* block.



Clustering in Graphs and Networks

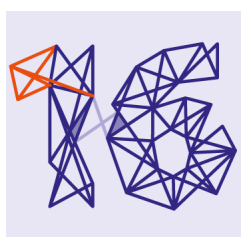
Since in a graph $\mathbf{G} = (V, L)$, we have two kinds of objects – nodes and links we can think about *clustering of nodes* and *clustering of links*. Usually we deal with clustering of nodes.

Again, we use the standard clustering methods – provided that we have an appropriate definition of dissimilarity between nodes. The usual approach is to define a vector description $[v] = [t_1, t_2, \ldots, t_m]$ of each node $v \in V$, and then use some standard dissimilarity, δ, on $\mathbb{R}^m$ to compare these vectors $d(u, v) = \delta([u], [v])$.

We can assign to each node v different neighborhoods, such as $N(v) = \{u \in V : (v, u) \in L\}$, and other sets. In these cases, the dissimilarities between sets are used on them.

For a given graph $\mathbf{G} = (V, L)$, a property $t : V \to \mathbb{R}$ is *structural* iff for every automorphism φ of $\mathbf{G}$ it holds

$$\forall v \in V : t(v) = t(\varphi(v))$$



Clustering Graphs and Networks indirect approach

Examples of such properties are:

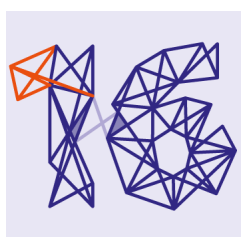
$t(v)$ = degree (the number of neighbors) of node v
$t(v)$ = the number of nodes at distance d from node v
$t(v)$ = the number of triads of type x at node v
$t(v)$ = the number of graphlets of type x at node v [Pržulj(2007)]

For a given graph $\mathbf{G} = (V, L)$, a *property of pairs of nodes* $q : V \times V \to \mathbb{R}$ is *structural* if for every automorphism φ of $\mathbf{G}$, it holds:

$$\forall u, v \in V : q(u, v) = q(\varphi(u), \varphi(v))$$

Some examples of structural properties of pairs of nodes are:

$q(u, v)$ = **if** $(u, v) \in L$ **then** 1 **else** 0;
$q(u, v)$ = number of common neighbors of units u and v;
$q(u, v)$ = length of the shortest path from u to v.



Clustering Graphs and Networks indirect approach

Using a selected property of pairs of nodes, q, we can describe each node u with a vector

$$[u] = [q(u, v_1), q(u, v_2), \ldots, q(u, v_n), q(v_1, u), \ldots, q(v_n, u)]$$

and we define the dissimilarity between nodes $u, v \in V$ as $d(u, v) = \delta([u], [v])$.

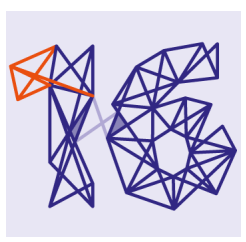
Corrected dissimilarities based on properties of pairs of nodes for measuring the similarity between nodes v_i and v_j ($p \geq 0$) must be used [Doreian, Batagelj, and Ferligoj(2005)] such as:

The corrected Manhattan distance:

$$d_c(p)(v_i, v_j) = \sum_{\substack{s=1 \\ s \neq i,j}}^{n} (|q_{is} - q_{js}| + |q_{si} - q_{sj}|) + p \cdot (|q_{ii} - q_{jj}| + |q_{ij} - q_{ji}|)$$

The corrected Euclidean distance:

$$d_e(p)(v_i, v_j) = \sqrt{\sum_{\substack{s=1 \\ s \neq i,j}}^{n} ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2) + p \cdot ((q_{ii} - q_{jj})^2 + (q_{ij} - q_{ji})^2)}$$



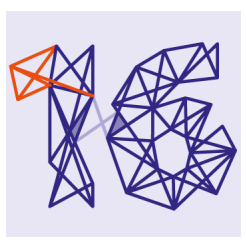
Clustering Graphs and Networks
A direct approach – blockmodeling

A partition $\mathbf{C} = \{C_i\}$ splits the set of links (arcs) $L \subseteq V \times V$ into *blocks* $B_{ij} = L \cap C_i \times C_j$ – a subgraph of arcs from cluster C_i to cluster C_j. In blockmodeling, analysts adopting this approach attempt to find partitions producing blocks of selected types (complete, empty, regular, etc.), while allowing for some 'errors' in the form of links not consistent with the specified block types [Doreian, Batagelj, and Ferligoj(2005)]. Usually, the relocation method is used for solving the corresponding optimization problems.

Not all clustering problems can be expressed by a simple criterion function. In some applications, a *general* criterion function of the form

$$P(\mathbf{C}) = \bigoplus_{(C_1,C_2)\in\mathbf{C}\times\mathbf{C}} q(C_1, C_2), \quad q(C_1, C_2) \geq 0$$

is needed. We use this in the optimizational approach to blockmodeling [Doreian, Batagelj, and Ferligoj(2005)].

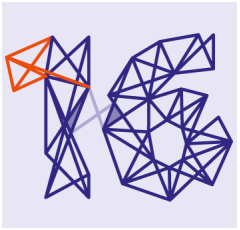


Example: Support network among informatics students

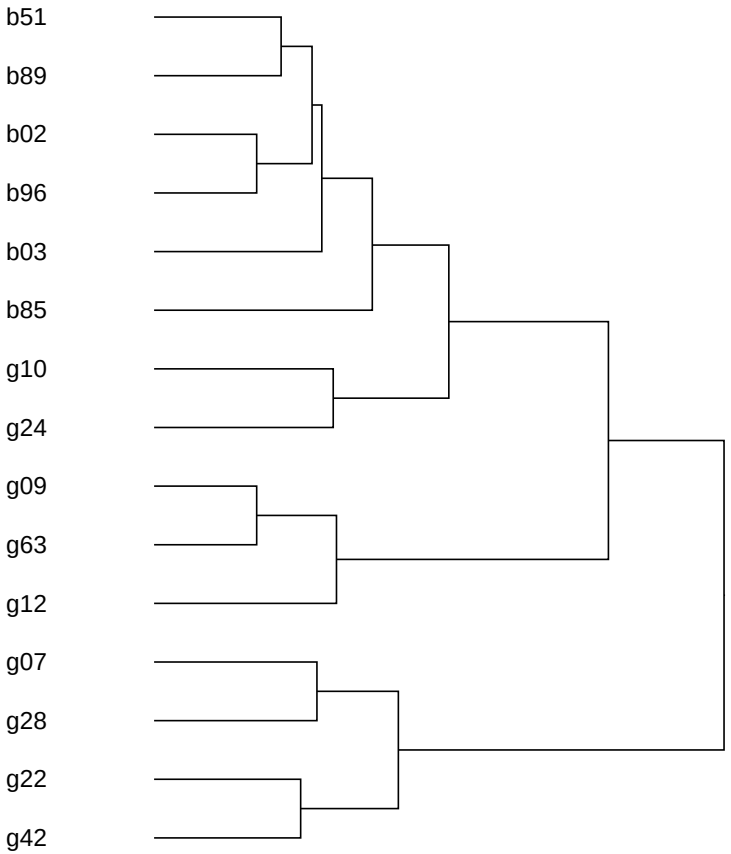
The analyzed network consists of social support exchange relation among fifteen students of the Social Science Informatics fourth year class (2002/2003) at the Faculty of Social Sciences, University of Ljubljana. Interviews were conducted in October 2002.
Support relation among students was identified by the following question:

> *Introduction: You have done several exams since you are in the second class now. Students usually borrow studying material from their colleagues.*
>
> *Enumerate (list) the names of your colleagues that you have most often borrowed studying material from. (The number of listed persons is not limited.)*

Nodes represent students in the class; circles – girls, squares – boys.
Opposite pairs of arcs are replaced by edges.

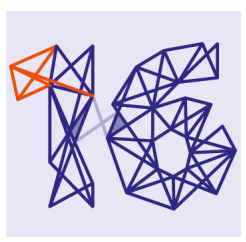


Indirect approach

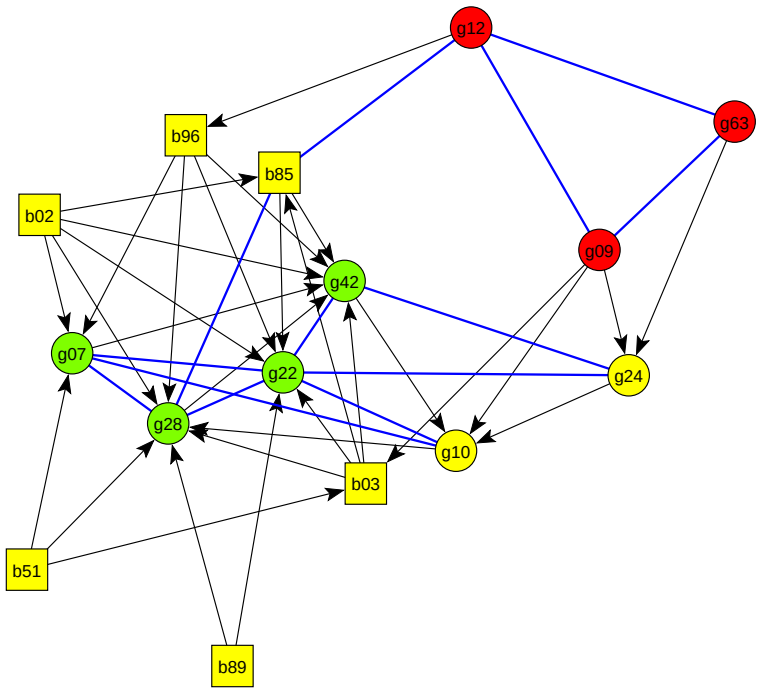


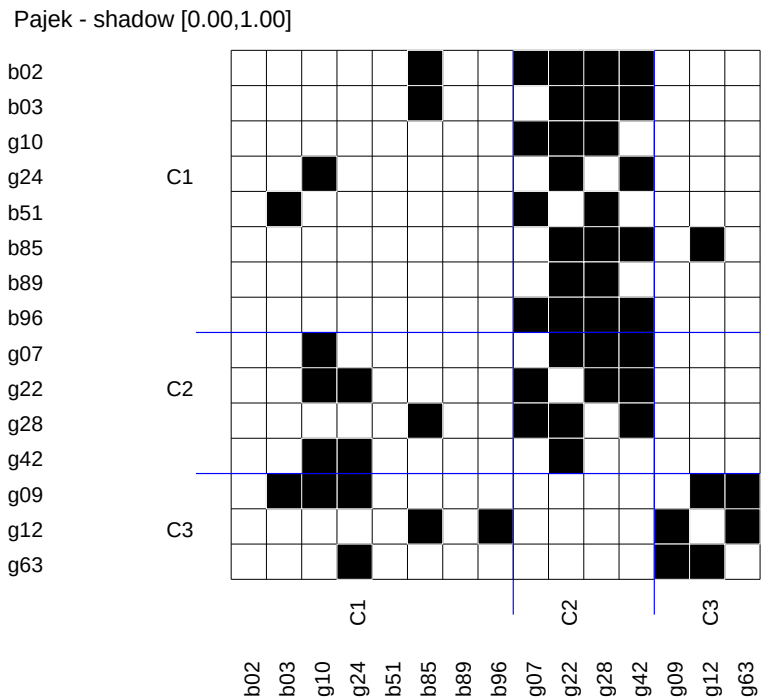
Using *Corrected Euclidean-like dissimilarity* and *Ward clustering method* we obtain the following dendrogram.
From it we can determine the number of clusters: 'Natural' clusterings correspond to clear 'jumps' in the dendrogram.
If we select 3 clusters we get the partition **C**.

$$\mathbf{C} = \{\{b51, b89, b02, b96, b03, b85, g10, g24\}, \{g09, g63, g12\}, \{g07, g28, g22, g42\}\}$$

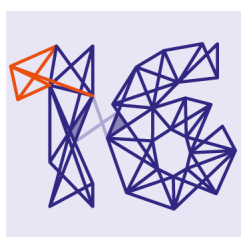


Partition in 3 clusters



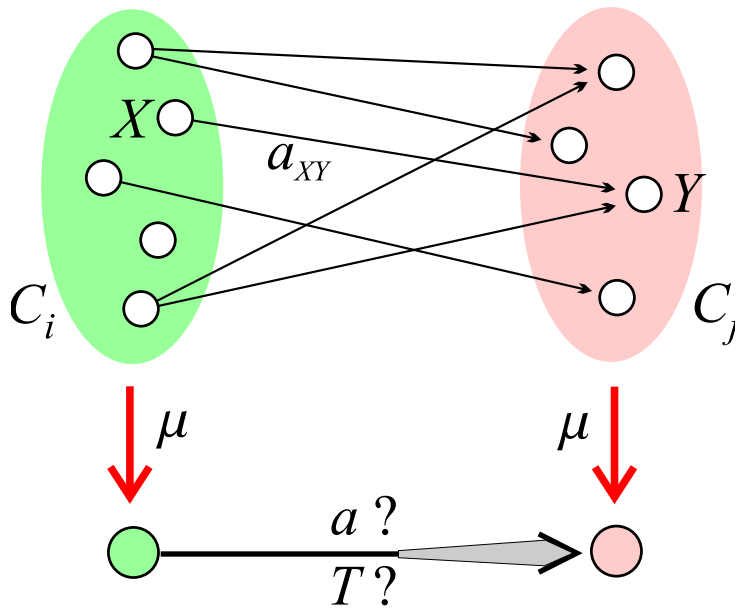


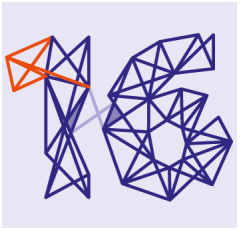
The partition can be used also to reorder rows and columns of the matrix representing the network. Clusters are divided using blue vertical and horizontal lines.



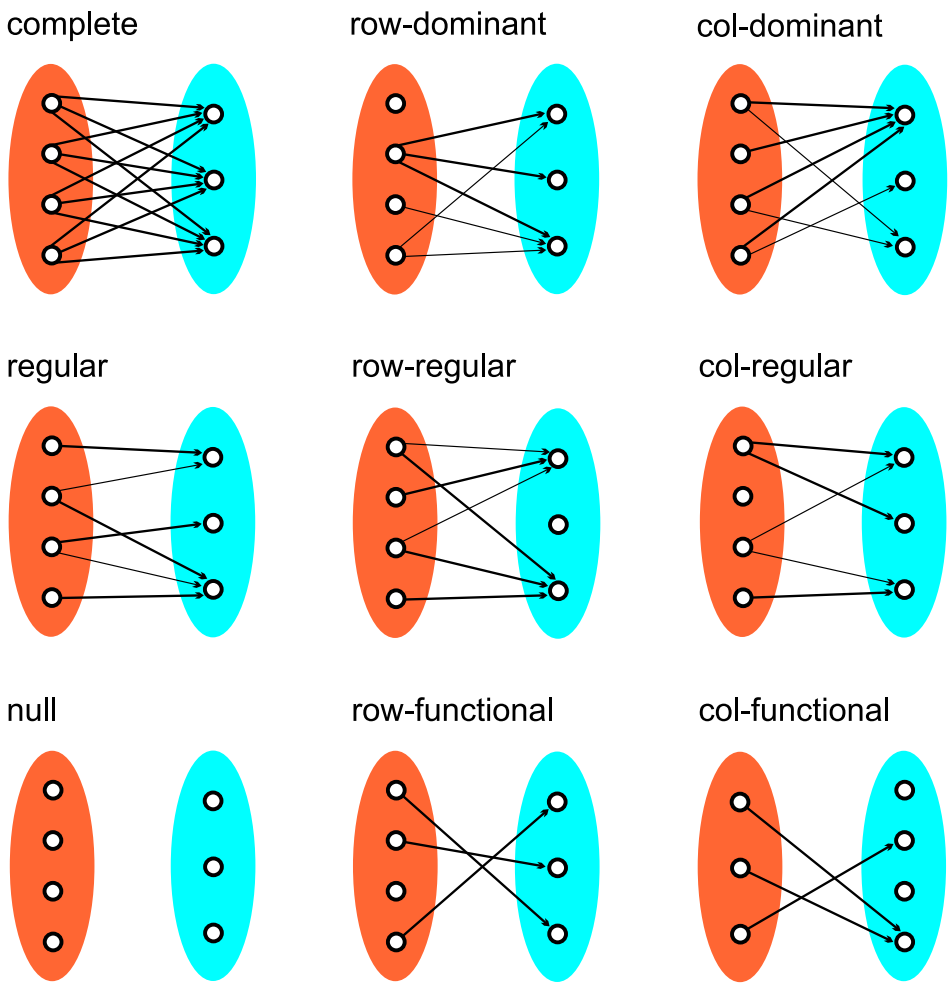
Generalized Blockmodeling

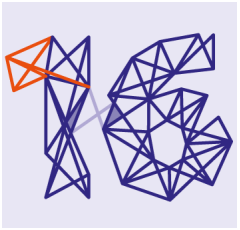
A *blockmodel* consists of structures obtained by identifying all units from the same cluster of the clustering **C**. For an exact definition of a blockmodel we have to be precise also about which blocks produce an arc in the *reduced graph* and which do not, and of what *type*. Some types of connections are presented in the figure on the next slide. The reduced graph can be represented by relational matrix, called also *image matrix*.





Block Types


complete
row-dominant
col-dominant
regular
row-regular
col-regular
null
row-functional
col-functional



Generalized equivalence / Block Types

	Y				
X	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1

complete

	Y				
X	0	1	0	0	0
	1	1	1	1	1
	0	0	0	0	0
	0	0	0	1	0

row-dominant

	Y				
X	0	0	1	0	0
	0	0	1	1	0
	1	1	1	0	0
	0	0	1	0	1

col-dominant

	Y				
X	0	1	0	0	0
	1	0	1	1	0
	0	0	1	0	1
	1	1	0	0	0

regular

	Y				
X	0	1	0	0	0
	0	1	1	0	0
	1	0	1	0	0
	0	1	0	0	1

row-regular

	Y				
X	0	1	0	1	0
	1	0	1	0	0
	1	1	0	1	1
	0	0	0	0	0

col-regular

	Y				
X	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0

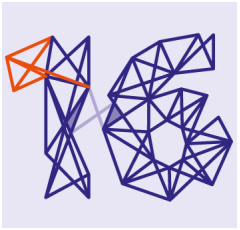
null

	Y				
X	0	0	0	1	0
	0	0	1	0	0
	1	0	0	0	0
	0	0	0	1	0

row-functional

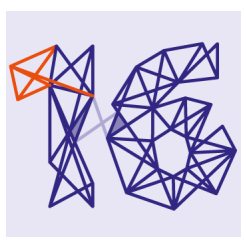
	Y				
X	1	0	0	0	
	0	1	0	0	
	0	0	1	0	
	0	0	0	0	
	0	0	0	1	

col-functional

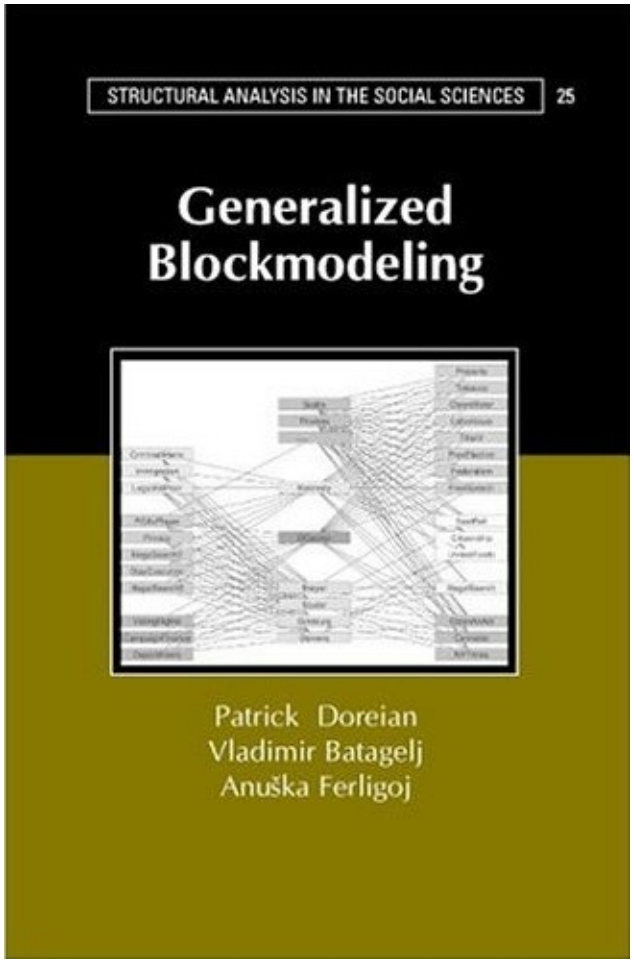


Conclusion

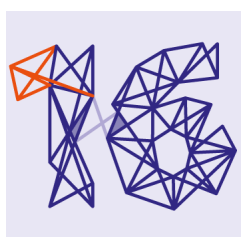
most of criterion functions are based on structural equivalence. One of the challenges for future research is to develop efficient algorithms for other types of equivalences for large networks.



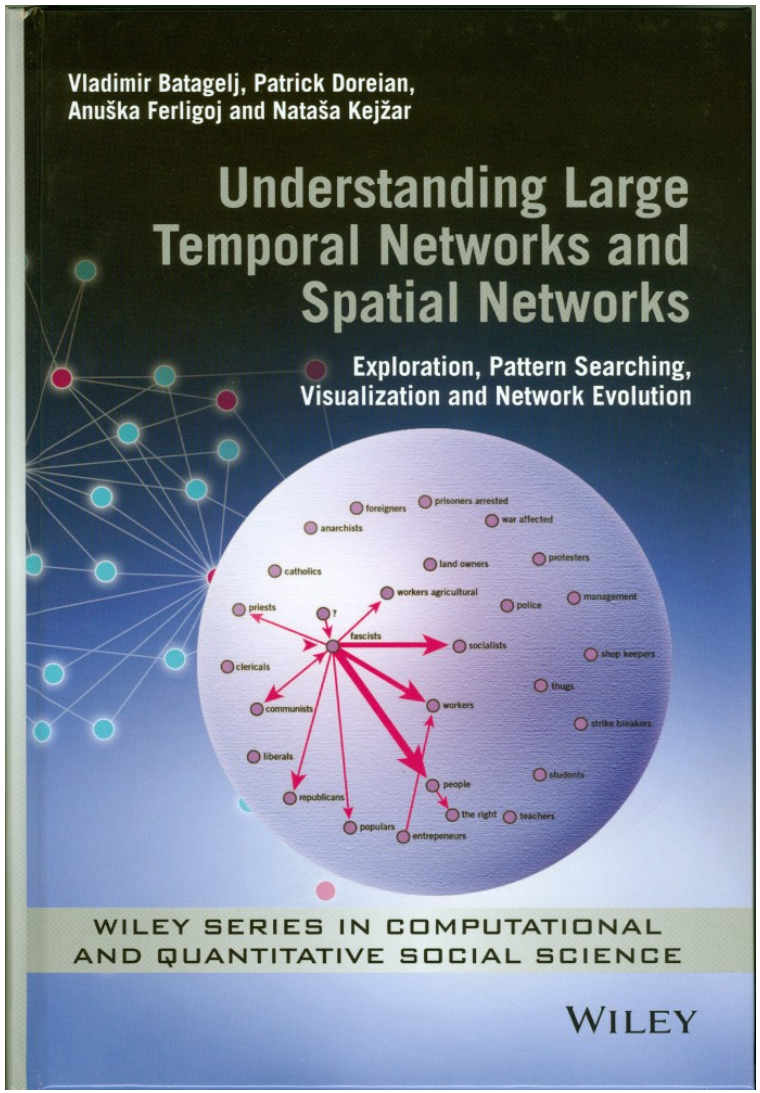
More on clustering and blockmodeling



The details about the clustering and (generalized) blockmodeling of networks can be found in our book:
P. Doreian, V. Batagelj, A. Ferligoj: *Generalized Blockmodeling*, CUP, 2005.

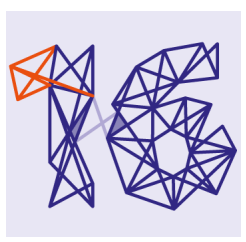


Understanding large networks



This lecture is closely related to chapters 2 and 3 in the book:

Vladimir Batagelj, Patrick Doreian, Anuška Ferligoj and Nataša Kejžar: Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution. Wiley Series in Computational and Quantitative Social Science. Wiley, October 2014.



References I

E. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.

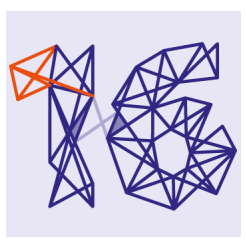
Ahmed, A., Batagelj, V., Fu, X., Hong, S.-H., Merrick, D., Mrvar, A.: Visualisation and analysis of the Internet movie database. Asia-Pacific Symposium on Visualisation 2007 (IEEE Cat. No. 07EX1615), 2007, p 17-24.

M. R. Anderberg. *Cluster Analysis for Application*. Academic Press, New York, 1973.

G. D. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(2):1–27, 2003.

Barabasi, A.L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A., Vicsek, T.: Evolution of the social network of scientic collaborations. Physica **311** (2002) 590–614

V. Batagelj. Note on ultrametric hierarchical clustering algorithms. *Psychometrika*, 46(3):351–352, 1981.



References II


V. Batagelj. Generalized Ward and related clustering problems. In H. Bock, editor, *Classification and Related Methods of Data Analysis*, pages 67–74. North-Holland, Amsterdam, 1988.

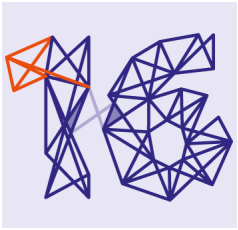

V. Batagelj. Similarity measures between structured objects. In A. Graovac, editor, *MATH/CHEM/COMP 1988: proceedings of an International Course and Conference on the Interfaces between Mathematics, Chemistry, and Computer Science, Dubrovnik, Yugoslavia, 20-25 June 1988*, Studies in physical and theoretical chemistry, pages 25–40. Elsevier, 1989.


Batagelj, V.: Wos2pajek – networks from web of science (2007). http://vladowiki.fmf.uni-lj.si/doku.php?id=pajek:wos2pajek


V. Batagelj. **clurc** – R package for clustering with relational constraint. 2017. URL `https://github.com/bavla/cluRC`.


Batagelj, V, Cerinšek, M: On bibliographic networks. Scientometrics 96 (2013) 3, 845-864.


V. Batagelj, S. Korenjak-Černe, and S. Klavžar. Dynamic programming and convex clustering. *Algorithmica*, 11(2):93–103, 1994.



References III

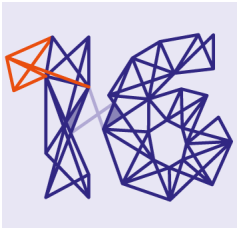
V. Batagelj and A. Ferligoj. Clustering relational data. In W. Gaul, O. Opitz, and M. Schader, editors, *Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 3–15. Springer, Berlin, Heidelberg, 2000.

Batagelj, V., Praprotnik, S.: An algebraic approach to temporal network analysis based on temporal quantities. Social Network Analysis and Mining, 6(2016)1, 1-22.

Batagelj, V., Zaveršnik, M.: Fast algorithms for determining (generalized) core groups in social networks. Advances in Data Analysis and Classification, 2011. Volume 5, Number 2, 129-145.

V. Batagelj, P. Doreian, A. Ferligoj, and N. Kejžar. *Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution*. Wiley Series in Computational and Quantitative Social Science Series. Wiley, 2014.

V. Batagelj, N. Kejžar, and S. Korenjak-Černe. Clustering of modal valued symbolic data. *arXiv preprint arXiv:1507.06683*, 2015.



References IV

J. Benzécri and L. Bellier. *L'analyse des données: La Taxinomie*, volume 1 of *L'analyse des données*. Dunod, 1973.

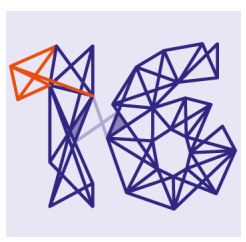
L. Billard and E. Diday. *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. Wiley Series in Computational Statistics. Wiley, 2012.

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

H.-H. Bock. A history of the international federation of classification societies. 2006. URL https://ifcs.boku.ac.at/site/lib/exe/fetch.php?media=pdfs:ifcs_history.pdf.

J. Bodlaj and V. Batagelj. Hierarchical link clustering algorithm in networks. *Physical Review E*, 91(6):062814, 2015.

P. Brucker. On the complexity of clustering problems. In R. Henn, B. Korte, and W. Oettli, editors, *Optimization and Operations Research*, volume 157 of *Lecture Notes in Economics and Mathematical Systems*, pages 45–54. Springer, Berlin, Heidelberg, 1978.



References V

M. Bruynooghe. Méthodes nouvelles en classification automatique des données taxinomiques nombreuses. *Statistique et Analyse des Données*, (3): 24–42, 1977.

D. Cartwright and F. Harary. Structural balance: A generalization of Heider's theory. *Psychological Review*, 63:277–293, 1956.

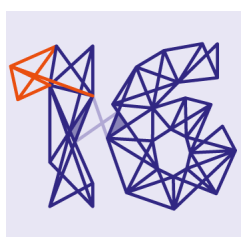
Cerinšek, M., Batagelj, V.: Network analysis of Zentralblatt MATH data. Scientometrics, 102(2015)1, 977-1001.

Cerinšek, M., Batagelj, V.: Generalized two-mode cores. Social Networks 42 (2015), 80–87.

T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction To Algorithms*. MIT Press, Cambridge, 2 edition, 2001.

E. Diday. *Optimisation en classification automatique, Tome 1., 2.* INRIA, Rocquencourt, 1979. (in French).

E. Diday and H. H. Bock. *Analysis of symbolic data: Exploratory methods for extracting statistical information from complex data*. Springer-Verlag, New York, 2000.



References VI

J. Dieudonné. *Foundations of modern analysis*. Academic Press, New York, 1960.

P. Doreian, V. Batagelj, and A. Ferligoj. *Generalized Blockmodeling*. Cambridge University Press, Cambridge, 2005.

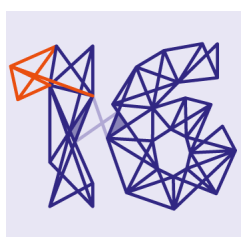
A.-H. Esfahanian. On the evolution of connectivity algorithms. In L. W. Beineke and R. J. Wilson, editors, *Topics in structural graph theory*, volume 147 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, New York, 2013.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. American Association for Artificial Intelligence Magazine (1996), 37–54

J. G. Fletcher, "A more general algorithm for computing closed semiring costs between vertices of a directed graph," *CACM* (1980), pp. 350-351.

A. Ferligoj and V. Batagelj. Some types of clustering with relational constraints. *Psychometrika*, 48(4):541–552, 1983.

A. Ferligoj and V. Batagelj. Clustering with relational constraint. *Psychometrika*, 47(4):413–426, 1982.



References VII

A. Ferligoj and L. Kronegger. Clustering of attribute and/or relational data. *Metodološki zvezki*, 6(2):135–153, 2009.

A. Ferligoj and V. Batagelj. Direct multicriteria clustering algorithms. *Journal of Classification*, 9:43–61, 1992.

G. Gan, C. Ma, and J. Wu. *Data Clustering – Theory, Algorithms, and Applications*. SIAM, Philadelphia, 2007.

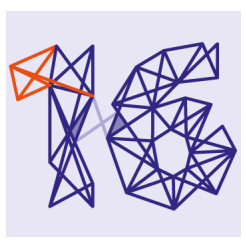
M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, 1979.

A. D. Gordon. *Classification, 2nd Edition*, volume 82 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, Boca Raton, 1999. ISBN 978-1584880134.

O. Grygorash, Y. Zhou, and Z. Jorgensen. Minimum spanning tree based clustering algorithms. In *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, pages 73–81. IEEE, 2006.

F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.

J. A. Hartigan. *Clustering algorithms*. Wiley-Interscience, New York, 1975.



References VIII

C. Hayashi. Chikio Hayashi and Data Science – What is data science? *Student*, 2(1):44–51, 1997.

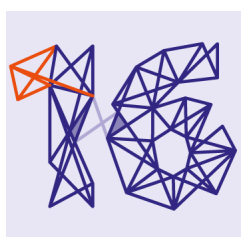
A. Jain and R. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988.

N. Jardine, P. Jardine, and R. Sibson. *Mathematical Taxonomy*. Wiley series in probability and mathematical statistics. Wiley, 1971.

S. D. Kamvar, D. Klein, and C. D. Manning. Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 283–290, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1): 359–392, 1998.

R. Kashyap and B. Oommen. A common basis for similarity measures involving two strings. *International Journal of Computer Mathematics*, 13 (1):17–40, 1983.



References IX

L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. A Wiley-Interscience publication. Wiley, 1990.

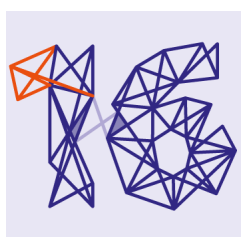
Kejžar, N., Korenjak Černe, S., Batagelj, V.: Network Analysis of Works on Clustering and Classification from Web of Science. Classification as a Tool for Research. Hermann Locarek-Junge, Claus Weihs eds. Proceedings of IFCS 2009. Studies in Classification, Data Analysis, and Knowledge Organization, 525-536, Springer, Berlin, 2010.

B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2):291–307, 1970.

S. Korenjak-Černe, N. Kejžar, and V. Batagelj. A weighted clustering of population pyramids for the world's countries, 1996, 2001, 2006. *Population studies*, 69(1):105–120, 2015.

D. Knuth. *The Stanford GraphBase, A Platform for Combinatorial Computing*. ACM Press, New York, 1993.

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965. English translation in Soviet Physics Doklady, 10(8):707-710, 1966.



References X

F. H. C. Marriott. Optimization methods of cluster analysis. *Biometrika*, 69 (2):417–421, 1982.

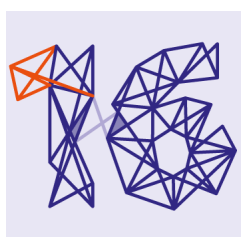
D. W. Matula. Graph theoretic techniques for cluster analysis algorithms. In J. Van Ryzin, editor, *Classification and clustering: proceedings of an advanced seminar conducted by the Mathematics Research Center, the University of Wisconsin-Madison, May 3-5, 1976*, pages 95–130. Academic Press, 1977.

J. Moreno. *Who shall survive? : a new approach to the problem of Human Interrelations*, volume 58 of *Nervous and mental disease monograph series*. Nervous and Mental Disease Publ., Washington, 1934.

F. Murtagh. *Multidimensional clustering algorithms*, volume 4. Physika Verlag, Vienna, 1985.

Newman, M.E.: The structure of scientific collaboration communities. Proceedings of the National Academy of Science (PNAS) **98** (2001) 404–409

M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.



References XI

W. D. Nooy, A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek, 3rd edition*. Cambridge University Press, New York, NY, USA, 2018.

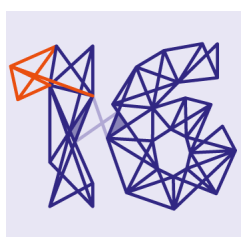
Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435** (2005) 814

Palla, G., Barabasi, A.L., Vicsek, T.: Quantifying social group evolution. Nature **446** (2007) 664-667

Perianes-Rodriguez, A., Waltman, L., Van Eck, N.J. (2016). Constructing bibliometric networks: A comparison between full and fractional counting. Journal of Informetrics, 10(4), 1178-1195.

N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

F. S. Roberts. *Discrete mathematical models, with applications to social, biological, and environmental problems*. Prentice-Hall, Englewood Cliffs, N.J., 1976.



References XII

R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy*. Books in biology. W. H. Freeman, 1963.

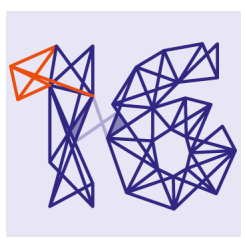
H. Späth. *Cluster-Analyse-Algorithmen: zur Objektklassifizierung und Datenreduktion*. Datenverarbeitung: Oldenbourg. Oldenbourg R. Verlag GmbH, 1977.

Wasserman, S., Faust, K.: Social network analysis: methods and applications. Cambridge Univ. Press, Cambridge, 1997.

Zaveršnik, M., Batagelj, V.: Islands. In: XXIV International Sunbelt Social Network Conference, Portorož, Slovenia (2004)

Pajek's wiki. http://pajek.imfm.si

Vladimir Batagelj, Andrej Mrvar: Pajek manual.



Acknowledgments

This work is supported in part by the Slovenian Research Agency (research program P1-0294 and research projects J1-9187 and J7-8279) and by Russian Academic Excellence Project '5-100'.